THE BOUNDARY METHOD AND GENERAL AUCTION FOR OPTIMAL MASS TRANSPORTATION AND WASSERSTEIN DISTANCE COMPUTATION

A Dissertation
Presented to
The Academic Faculty

By

Joseph Donald Walsh III

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Mathematics

Georgia Institute of Technology

August 2017

THE BOUNDARY METHOD AND GENERAL AUCTION FOR OPTIMAL MASS TRANSPORTATION AND WASSERSTEIN DISTANCE COMPUTATION

Approved by:

Dr. Luca Dieci, Advisor
School of Mathematics
Georgia Institute of Technology

Dr. Hao-Min Zhou
School of Mathematics
Georgia Institute of Technology

Dr. Sung Ha Kang
School of Mathematics
Georgia Institute of Technology

Dr. Michael Muskulus
Department of Civil
and Environmental Engineering
Norwegian University of Science and Technology

Dr. Anthony Yezzi
School of Electrical
and Computer Engineering
Georgia Institute of Technology

Date Approved: April 4, 2017

Dilegua, o notte!
Tramontate, stelle!
Tramontate, stelle!
All'alba vincerò!
vincerò, vincerò!

Giacomo Puccini, composer
Giuseppe Adami and Renato Simoni, librettists

To Jo, Luca, and Sandy.

ACKNOWLEDGEMENTS

None of what appears here would have been possible without Luca Dieci's sage advice, steady guidance, and occasionally frustrating insistence that the details be just right. Luca saw my potential before we even met, taking a chance on a nontraditional student from a small midwestern university. I was constantly reassured and gratified by his comfortable patience in the face of my own intensity and agitation. I could not have asked for a better mentor or more pleasant companion on this journey.

My research would not have been possible without the support of John Festa and the National Science Foundation.[1] Through the Graduate Research Fellowship, the NSF gave me three years to hone my research ideas. Along with his financial support, John gave an even more precious gift: his time, spent advising me on my life and future prospects. I appreciate both gifts.

The research of Michael Muskulus inspired my own, and our semester-long collaboration introduced me to many of the ideas and approaches that form the backbone of this thesis. Wherever I end up, I know he set me on the right track.

Sung Ha Kang never hesitates to tell me when I can do better, and always takes the time to help me understand how. If any part of this thesis approaches perfection, it is due to her influence.

Hao-Min Zhou and Wilfrid Gangbo opened my eyes to the potential of numerical optimal transport. I was swimming in the shallows before my conversations with them showed me the deep waters all around. Seeing optimal transport from their perspective greatly aided my own understanding.

As part of his teaching duties, Anthony Yezzi helped me learn about applications of numerical partial differential equations that formed a key element of my doctoral minor. He

[1]This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1650044. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

went above and beyond that, serving on my doctoral committee and providing important feedback about potential applications of my research.

Pavel Bochev generously gave his time to read about my research and question me about its impact. In his role as a journal editor, he also gave me the opportunity to apply my expertise to the evaluation of new work in my field. By doing so, he helped me refine the Background and Conclusions of this work, integrating everything into a more coherent whole.

Dimitri Bertsekas and David Castañón took time out of their busy schedules to answer the questions of a curious graduate student trying to do their work justice. I hope I did. I know this thesis is better for their kind advice.

John Etnyre, Mohammed Ghomi, and Klara Grodzinsky were instrumental in making the School of Mathematics a welcoming environment and in sharing what I needed to know to become, not just a skilled researcher, but a well-rounded academician. I learned much from their example of teaching, service, and collegiality.

No list of acknowledgements would be complete without thanking the undergraduate instructors who did the most to inspire my love of mathematics, setting me on this path: Sister Ann Mason of Aquinas College, and Tim McGrew, Jay Treiman, and Art White of Western Michigan University. Sr. Ann taught me the joy of calculus, and Professor Treiman the joy of analysis. Tim showed me the mathematics in philosophy, and Art the philosophy in mathematics. I hope I have done justice to their example.

Finally, any acknowledgement of credit must include Jo: my rock, my compass, and my partner. Without her, I literally would not be here.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Numerical optimal transport is an important area of research, but most problems are too large and complex for easy computation. Because continuous transport problems are generally solved by conversion to either discrete or semi-discrete forms, I focused on methods for those two.

I developed a discrete algorithm specifically for fast approximation with controlled error bounds: the general auction method. It works directly on real-valued transport problems, with guaranteed termination and *a priori* error bounds.

I also developed the boundary method for semi-discrete transport. It works on unaltered ground cost functions, rapidly identifying locations in the continuous space where transport destinations change. Because the method computes over region boundaries, rather than the entire continuous space, it reduces the effective dimension of the discretization.

The general auction is the first relaxation method designed for compatibility with real-valued costs and weights. The boundary method is the first transport technique designed explicitly around the semi-discrete problem and the first to use the shift characterization to reduce dimensionality. No truly comparable methods exist.

The general auction and boundary method are able to solve many transport problems that are intractable using other approaches. Even where other solution methods exist, my tests indicate that the general auction and boundary method outperform them.

CHAPTER 1
INTRODUCTION

Mass transport is the process of moving density, or "weight," from one metric space to another. The expense of moving these weights is computed using a ground cost function. The total expense over the entire product space is the transport cost. Optimal transport is the process of determining the lowest possible transport cost. Optimal transport costs constitute a distance, known as the Wasserstein distance.

Computing optimal mass transport is important. Transport equations have promising applications in medicine, economics, computer science, physics, and multiple engineering disciplines. For this reason, numerical approaches to the transport problem have received considerable attention over the last thirty years. Active, dedicated research groups are working in multiple countries.

Solutions to transport problems are difficult to compute. When attacked naively, computation is slow, expensive, and unstable. To date, computational successes have been largely restricted to problems with well-behaved metrics and specialized cost functions, either the squared or regularized Euclidean distance. Even the simplest three-dimensional problems require high-powered equipment and specialized techniques. Specialized approaches to the discrete problem have been shown to scale super-cubically with respect to the number of nodes. Specialized approaches to the continuous problem must satisfy restrictive well-posedness conditions, requiring regularization for most applications, and they still appear to scale quadratically or worse with respect to the total size of the discretization.

Key challenges identified in numerical optimal transport are:

(a) numerical methods capable of handling general ground costs,

(b) efficient computation of Wasserstein distances, and

(c) general techniques for solving three (or higher) dimensional problems.

Whether or not a transport problem is continuous, current approaches discretize at least one space, solving problems that are either semi-discrete or fully discrete. For that reason, I concentrated my research on discrete and semi-discrete optimal transport.

I created the general auction for discrete transport. Auction algorithms are intuitively based on real-world sales auctions, but they depended on integral assignment problems. Without integral ground costs, convergence is not guaranteed, and without integral weights, computation cannot be performed at all. When weights are not unitary, the problem must be expanded, greatly impacting speed and storage. My general auction algorithm is based directly on the real-valued transport problem, and guarantees convergence for such data. It allows one to quickly and accurately approximate solutions for optimal transport, particularly when the ground cost is a distance.

I also created the boundary method for semi-discrete transport. By an innovative application of the shift characterization developed by Rüschendorf and Uckelmann, I am able to solve problems with arbitrary cost functions. Rather than approximate the entire transport plan, the boundary method identifies region boundaries and uses relationships between the adjacent regions to obtain the necessary information. This reduces the dimension of the transport problem, enabling faster computation.

As I will show, both methods are able to handle general ground costs. They efficiently compute the Wasserstein distance when it has a closed-form expression. They have also been applied successfully to both two and three-dimensional problems, and display every indication of generalizing to arbitrarily high dimensions. Even with complicated irrational data, during testing the scaling of the general auction was no worse than $\mathcal{O}(N^3)$, where N is the number of nodes. The boundary method was able to solve semi-discrete problems in $\mathbb{R}^d \times \mathbb{R}^d$ with complexity scaling equal to $\mathcal{O}(W^{d-1} \log W)$, where W is the width of the discretization. Since the total size of the discretization is $N = W^d$, this is equivalent to $\mathcal{O}(N^{1-1/d} \log N)$.

Chapters 2 and 3 provide background from the transport literature. In Chapter 2, I describe the continuous and discrete transport problems in detail, give a brief overview of existing analytical results and numerical methods, and describe applications receiving active research. Chapter 3 gives a detailed description of the auction methods for optimal transport that are described in the literature.

Chapters 4 and 5 present my original work and demonstrate its significance. In Chapter 4, I explain the general auction method I created and provide mathematical support, including proof of convergence for real-valued costs and weights. I also offer numerical results, comparing the general auction to existing methods for integral-valued data, and providing scaling results when the general auction is applied to real-valued data.

Chapter 5 begins by defining the semi-discrete transport problem and describing the boundary method. It supports the method with mathematical proof, including proof of convergence with respect to the Wasserstein distance. Extensive numerical results show the generality and overall effectiveness of the method.

Finally, Chapter 6 offers conclusions based on the results I obtained. It summarizes what has been accomplished, describes what remains to be done, and outlines some directions for future research.

CHAPTER 2
BACKGROUND

The theory of optimal transport dates back to the work by Monge in 1781 [86]. In the 1940s, Kantorovich's papers [71, 72] relaxed Monge's requirement that no mass be split, creating what is now known as the Monge-Kantorovich problem.

Transport problems are characterized in two fundamentally different ways, based on whether the problem is continuous or discrete. The continuous problem must be embedded in two metric spaces $(X,\, \mu)$ and $(Y,\, \nu)$, and there must be a ground cost function establishing a distance between every pair of points $\mathbf{x} \in X$ and $\mathbf{y} \in Y$. The discrete problem requires no such embedding, and not every pair needs to be related.

The two problems are not directly related: not every problem is continuous, and not every discrete problem relates all points in the context of a complete embedding. However, the two can be related — for example, when a continuous problem is discretized — so there are distinct advantages to considering both. I describe continuous and discrete transport as distinct, separate entities, but Remark 2.2.1 considers how they may be related.

2.1 The continuous transport problem

I use Kantorovich's relaxed problem as the basis for the definition of continuous optimal transport.

2.1.1 The Monge-Kantorovich problem

Definition 2.1.1 (Monge-Kantorovich)**.** Let $X,\, Y \subseteq \mathbb{R}^d$, and let μ and ν be probability densities defined on X and Y, and let $c(\mathbf{x},\, \mathbf{y}) : X \times Y \to \mathbb{R}$ be a measurable *ground cost*

function. Define the set of *transport plans*

$$\Pi(\mu,\,\nu) := \left\{ \pi \in \mathcal{P}(X \times Y) \;\middle|\; \begin{array}{c} \pi[A \times Y] = \mu[A],\, \pi[X \times B] = \nu[B]\,, \\ \forall \text{ meas. } A \subseteq X,\, B \subseteq Y \end{array} \right\}, \tag{2.1.1}$$

where $\mathcal{P}(X \times Y)$ is the set of probability measures on the product space. Define the *primal cost* function $P : \Pi(\mu,\,\nu) \to \mathbb{R}$ as

$$P(\pi) := \int_{X \times Y} c(\mathbf{x},\,\mathbf{y})\, d\pi(\mathbf{x},\,\mathbf{y}). \tag{2.1.2}$$

The Monge-Kantorovich problem is to find the *optimal primal cost*

$$P^* := \inf_{\pi \in \Pi(\mu,\,\nu)} P(\pi), \tag{2.1.3}$$

and an associated *optimal transport plan*

$$\pi^* := \arg\inf_{\pi \in \Pi(\mu,\,\nu)} P(\pi). \tag{2.1.4}$$

2.1.2 The dual Monge-Kantorovich problem

Kantorovich also identified the problem's dual formulation.

Definition 2.1.2 (Dual Monge-Kantorovich)**.** Define the set of functions

$$\Phi_c(\mu,\,\nu) := \left\{ (\varphi,\,\psi) \in L^1(d\mu) \times L^1(d\nu) \;\middle|\; \begin{array}{c} \varphi(\mathbf{x}) + \psi(\mathbf{y}) \leq c(\mathbf{x},\,\mathbf{y})\,, \\ d\mu \text{ a.e. } \mathbf{x} \in X,\, d\nu \text{ a.e. } \mathbf{y} \in Y \end{array} \right\}. \tag{2.1.5}$$

Let the *dual cost* function, $D : \Phi_c(\mu,\,\nu) \to \mathbb{R}$, be defined as

$$D(\varphi,\,\psi) := \int_X \varphi\, d\mu + \int_Y \psi\, d\nu. \tag{2.1.6}$$

Then, the *optimal dual cost* is

$$D^* := \sup_{(\varphi,\,\psi)\in\Phi_c(\mu,\,\nu)} D(\varphi,\,\psi), \tag{2.1.7}$$

and an optimal dual pair is given by

$$(\varphi^*,\,\psi^*) := \arg\sup_{(\varphi,\,\psi)\in\Phi_c(\mu,\,\nu)} D(\varphi,\,\psi). \tag{2.1.8}$$

For a given ground cost function c, solutions to Monge-Kantorovich problems are related to the *Wasserstein metric*, a distance between probability distributions:

$$W_p(\mu,\,\nu) := \inf_{\pi\in\Pi(\mu,\,\nu)} \left(\int_{X\times Y} c(\mathbf{x},\,\mathbf{y})^p\, d\pi(\mathbf{x},\,\mathbf{y})\right)^{1/p}. \tag{2.1.9}$$

For any given μ, ν, and c, we have $W_1(\mu,\,\nu) = P^* = D^*$. Hence, we may refer to any of these as the *Wasserstein distance*, the *optimal transport cost*, or simply the *optimal cost*.

Remark 2.1.3. $W_p(\mu,\,\nu)$ is often written as W_p, with μ and ν implied. Furthermore, as Equation (2.1.9) makes clear, $W_p(\mu,\,\nu)$ always depends on the ground cost function $c(\mathbf{x},\,\mathbf{y})$. In the literature, W_1 is sometimes used as a default notation when the ground cost is given by the Euclidean distance $\|\mathbf{x}-\mathbf{y}\|_2$ (e.g., see [76]), and W_2^2 is occasionally used to indicate that the ground cost is the squared-Euclidean distance $\|\mathbf{x}-\mathbf{y}\|_2^2$.

2.1.3 The Monge problem

Definition 2.1.4 (Monge)**.** In certain cases, there exists at least one solution to the semi-discrete Monge-Kantorovich problem that does not split transported masses. In other words, there exists some π^* such that

$$\pi^*(\mathbf{x},\,\mathbf{y}) = \pi^*_{T^*}(\mathbf{x},\,\mathbf{y}) := \mu(\mathbf{x})\,\delta[\mathbf{y} = T^*(\mathbf{x})], \tag{2.1.10}$$

where $T^* : X \to Y$ is a measurable map, which we call the optimal transport map. When such a π^* exists, we say that the solution also satisfies the Monge problem.

If the Monge problem has a solution, we can assume without loss of generality that every $\pi \in \Pi(\mu, \nu)$ satisfies

$$\pi(\mathbf{x}, \mathbf{y}) = \pi_T(\mathbf{x}, \mathbf{y}) := \mu(\mathbf{x})\, \delta[\mathbf{y} = T(\mathbf{x})], \tag{2.1.11}$$

for some measurable transport map $T : X \to Y$, and the primal cost can be written as

$$P(\pi) := \int_X c(\mathbf{x}, T(\mathbf{x}))\, d\mu(\mathbf{x}). \tag{2.1.12}$$

2.2 The discrete transport problem

The discrete transport problem is a linear programming problem, and so it has a dual formulation and a complementary slackness condition. These are both given below.

It is possible to define the discrete transport problem as a special case of the problem given in Section 2.1. However, there are advantages to defining a different, more general discrete problem. In particular, describing discrete optimal transport as a maximization problem improves compatibility with the form used in the numerical literature. We will consider a distinct discrete problem, defined below, which complements the continuous form defined in Section 2.1. Remark 2.2.1 describes how the two problems are related.

2.2.1 Discrete transport

Consider the discrete transport problem, $\mathcal{T}$ which we will define as follows:

Suppose we are given a *demand* vector $\{d_i\}_{i=1}^M$ and a *supply* vector $\{s_j\}_{j=1}^N$, whose demand coefficients d_i and supply coefficients s_j are positive scalars such that

$$\sum_{i=1}^{M} d_i = \sum_{j=1}^{N} s_j = L > 0. \tag{2.2.1}$$

We refer to L as the *total weight* of the transport problem. In the underlying transport graph, the vertex i associated with demand coefficient d_i is a *sink*, and the vertex j associated with supply coefficient s_j is a *source*.

Furthermore, suppose for each i we are given the nonempty set

$$A(i) := \{\, j \in \mathbb{N}_n \mid \exists \text{ an arc connecting } j \text{ to } i \,\}. \tag{2.2.2}$$

Then the set of all possible transport pairs is equal to

$$\mathcal{A} := \{\, (i,\, j) \mid j \in A(i),\, i = 1,\, \ldots,\, n \,\}. \tag{2.2.3}$$

Thus, $\mathcal{A}$ is the set of arcs of the underlying transport graph, which has $M + N$ vertices and $|\mathcal{A}| \leq MN$ arcs. Without loss of generality, assume $M \geq N$.

For each $(i,\, j) \in \mathcal{A}$, let $c_{ij} < 0$ be the *cost coefficient* (or simply *cost*). Our goal is to

$$\begin{aligned}
&\text{maximize} && \sum_{(i,\, j) \in \mathcal{A}} c_{ij} f_{ij} && && \text{(2.2.4a)}\\
&\text{subject to} && \sum_{\{\, j \mid j \in A(i) \,\}} f_{ij} = d_i && \forall\, i = 1,\, \ldots,\, M, && \text{(2.2.4b)}\\
& && \sum_{\{\, i \mid j \in A(i) \,\}} f_{ij} = s_j && \forall\, j = 1,\, \ldots,\, N, && \text{(2.2.4c)}\\
& && 0 \leq f_{ij} \leq \min\{d_i,\, s_j\} && \forall\, (i,\, j) \in \mathcal{A}. && \text{(2.2.4d)}
\end{aligned}$$

We refer to f_{ij} as the *flow* along $(i,\, j)$.

Notice that we have assumed negative costs and formulated the transport problem as a maximization problem. This reflects the usual implementation of the auction method. Because $c_{ij} < 0$, the maximization equation, Equation (2.2.4a), provides a minimum overall

cost (by reversing the signs on c_{ij}). We refer to

$$\sum_{(i,\,j)\in\mathcal{A}} c_{ij} f_{ij} \tag{2.2.5}$$

as the *primal cost*. The solution to Equation (2.2.4a) is called the *optimal primal cost*, or *optimal cost*, of the transport problem, and is denoted by P^*.

2.2.2 Transport plan

A *transport plan* (or *map*) T is a multiset of triples $(i,\, j,\, q_{ij})$ such that the arc $(i,\, j) \in \mathcal{A}$ and the transported *quantity* q_{ij} is non-negative. Note that T may be empty. While the elements of T are not necessarily unique, for each $(i,\, j)$ we can compute the flow f_{ij} given by T as

$$f_{ij} := \sum_{\{\,(k,\,l,\,q_{kl})\in T \mid (k,\,l)=(i,\,j)\,\}} q_{kl}. \tag{2.2.6}$$

In order to apply the plan to our transport problem, we require that T satisfies $f_{ij} \leq \min\{d_i,\, s_j\}$ for all $(i,\, j) \in \mathcal{A}$. By a minor abuse of notation, we may say $(i,\, j) \in T$ to signify that $(i,\, j,\, q_{ij}) \in T$ for some $q_{ij} > 0$. We may also say $T \in \mathcal{T}$ to refer to some transport plan T associated with the transport problem $\mathcal{T}$.

Given any transport plan T, we say that sink i is *satisfied* if

$$\sum_{\{\,q_{ij} \mid (i,\,j,\,q_{ij})\in T\,\}} q_{ij} = d_i. \tag{2.2.7}$$

Otherwise, we say that i is *unsatisfied*. (Alternately, we may say i has unsatisfied demand D_i, where $0 \leq D_i \leq d_i$.)

Similarly, when

$$\sum_{\{\,q_{ij} \mid (i,\,j,\,q_{ij})\in T\,\}} q_{ij} = s_j. \tag{2.2.8}$$

we say the source j is *unavailable*. Otherwise, we say that j is *available*, or that j has available supply S_j, where $0 \leq S_j \leq s_j$.

A transport plan is said to be *feasible* or *complete* when all sinks are satisfied; otherwise the plan is called *partial*.

If T is such that the pair $(i,\ j)$ appears at most once, and $(i,\ j,\ q_{ij}) \in T$ implies $q_{ij} > 0$, we refer to T as a *simplified* transport plan. In this case $q_{ij} = f_{ij}$, and we may refer to flow and quantity interchangeably. Simplified transport plans greatly improve clarity of notation, so even though it is not strictly necessary, we will often assume T is simplified when stating definitions and proofs.

2.2.3 Dual problem

The dual transport problem can be written as

$$\min_{u_i,\, p_j} \left\{ \sum_{i=1}^{M} d_i u_i + \sum_{j=1}^{N} s_j p_j \right\} \tag{2.2.9}$$

with the restriction that $u_i + p_j \leq c_{ij}$ for all $(i,\ j) \in \mathcal{A}$. As usual, the dual variable u_i is called a *slack variable*, and the dual variable p_j is called a *price* of j. We call the vector $p = \{p_j\}_{j=1}^{N}$ a price vector of $\mathcal{T}$. Assume $p_j \geq 0$ for all j.

Given some price vector p, the *total expense* (or simply *expense*) associated with source $j \in A(i)$ for sink i is given by

$$x_{ij} = c_{ij} - p_j \tag{2.2.10}$$

and the *expense* for the sink i is

$$x_i = \max_{j \in A(i)} x_{ij}. \tag{2.2.11}$$

Because $c_{ij} < 0$, the maximization of the expense x_i actually generates the least overall expense (found by reversing the sign of x_i).

Suppose we have a simplified complete transport plan T and a price vector p. From linear programming theory, we know $(T,\, p)$ is simultaneously primal and dual optimal if and only if

$$u_i = \max_{k \in A(i)} \{c_{ik} - p_k\} = c_{ij} - p_j \qquad \forall\, (i,\, j,\, f_{ij}) \in T. \tag{2.2.12}$$

In other words, the loss to each sink is minimized by transport to the least expensive source(s). This is known as the *complementary slackness condition*, or *complementary slackness*.

Among other things, complementary slackness implies that the dual problem can only be minimized when $u_i = x_i$ for all i. Thus, we can view the prices p_j as the only variables in our dual problem, which we can rewrite as

$$\min_{p=\{p_j\}_{j=1}^{N}} \left\{ \sum_{i=1}^{M} d_i \max_{j \in A(i)} \{c_{ij} - p_j\} + \sum_{j=1}^{N} s_j p_j \right\} \tag{2.2.13}$$

Because $c_{ij} < 0$, the minimization equation, Equation (2.2.13), provides a maximum overall profit (by reversing the signs on c_{ij}). We refer to

$$\sum_{i=1}^{M} d_i \max_{j \in A(i)} \{c_{ij} - p_j\} + \sum_{j=1}^{N} s_j p_j \tag{2.2.14}$$

as the *dual profit*. The solution, given by Equation (2.2.13), is called the *optimal dual profit*, or *optimal profit*, of the transport problem, and is denoted by D^*.

Remark 2.2.1. The continuous and discrete problems defined above do not always map to one another, but it is possible to define conditions under which a mapping between formulations is possible.

- Suppose a problem is formulated as described in Section 2.1, with the added condition that μ and ν are discrete. For each $\mathbf{x}_i \in X$ such that $\mu(\mathbf{x}_i) = d_i > 0$, and $\mathbf{y}_j \in Y$ such that $\nu(\mathbf{y}_j) = s_j > 0$, let $c_{ij} = -c(\mathbf{x}_i,\, \mathbf{y}_j)$.

- Suppose instead that a discrete problem is formulated as given in Section 2.2.1, and $(i,\, j) \in \mathcal{A}$ for all $i \in \mathbb{N}_M$, $j \in \mathbb{N}_N$. There must also exist an embedding in $\mathbb{R}^d \times \mathbb{R}^d$ and a ground cost function c such that if $\mathbf{x}_i \in X$ has $\mu(\mathbf{x}_i) = d_i > 0$ and $\mathbf{y}_j \in Y$ has $\nu(\mathbf{y}_j) = s_j > 0$, then $c(\mathbf{x}_i,\, \mathbf{y}_j) = -c_{ij}$.

For both of these mappings, any optimal transport map for the continuous formulation is an optimal map under the discrete formulation, and the optimal costs differ only by a sign.

2.3 Analytical background

Monge proposed the discrete transport problem in the eighteenth century [86]. In the 1940s, Kantorovich extended the problem to continuous transport [71, 72]. A 1987 note by Yann Brenier, *Décomposition polaire et réarrangement monotone des champs de vecteurs*, linked transport problems to certain partial differential equations [24].

First and foremost of these partial differential equations is the Monge-Ampère equation:

$$|\nabla^2 u| = f, \tag{2.3.1}$$

where u is a smooth function. If f is Borel measurable, then a convex function $u \in C(\Omega)$ is a generalized, or Aleksandrov, solution if the Monge-Ampère measure Mu associated with the function u equals f. The general definition of Mu can be difficult to work with, but if we assume further that $u \in C^2(\Omega)$ is a convex function, then the Monge-Ampère measure satisfies

$$Mu(E) := \int_E |\nabla^2 u(\mathbf{z})|\, d\mathbf{z} \tag{2.3.2}$$

for all Borel sets $E \subset \Omega$. See [65] for more details.

The Monge-Kantorovich dual problem for the Euclidean distance can be reformulated as a Monge-Ampère-type partial differential equation [49]:

$$-\nabla \cdot (a \nabla u) = f, \text{ where } |\nabla u| \leq 1,\, a \geq 0, \text{ and } |\nabla u| < 1 \implies a = 0. \tag{2.3.3}$$

Generally speaking, the solution to an optimal transport problem may not be unique. A standard example of this is described in [115]:

> Suppose that we have n books of equal width on a shelf (the real line), arranged in a single contiguous block. We wish to rearrange them into another contiguous block, but shifted one book-width to the right. Two obvious candidates for the optimal transport plan present themselves:
>
> 1. move all n books one book-width to the right; ("many small moves")
> 2. move the left-most book n book-widths to the right and leave all other books fixed. ("one big move")
>
> If the cost function is proportional to Euclidean distance ($c(\mathbf{x}, \mathbf{y}) = \alpha|\mathbf{x} - \mathbf{y}|$) then these two candidates are both optimal. If, on the other hand, we choose the strictly convex cost function proportional to the square of Euclidean distance ($c(\mathbf{x}, \mathbf{y}) = \alpha|\mathbf{x} - \mathbf{y}|^2$), then the "many small moves" option becomes the unique minimizer.

Hence, it becomes vitally important to determine the conditions under which the partial differential equation is well-posed, in the sense of Hadamard. These conditions are understood to be the Ma-Trudinger-Wang (MTW) conditions, described in [81].

Most functions, particularly norms, do not satisfy the MTW conditions. Two classes of functions that do are *strictly convex functions*, such as the squared Euclidean distance used in the quotation, and *c-convex functions*. One way of defining c-convexity is the following (from [102]):

Definition 2.3.1 (c-convex function)**.** A function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is called *c-convex* if:

$$\exists\, \mathbf{x} \quad \text{s.t.} \quad f(\mathbf{x}) < +\infty, \tag{2.3.4}$$

$$\forall\, \mathbf{x} \qquad\quad f(\mathbf{x}) > -\infty, \tag{2.3.5}$$

and f has a representation of the form

$$f(\mathbf{x}) := \sup_{\mathbf{y} \in Y} \{c(\mathbf{x}, \mathbf{y}) + a(\mathbf{y})\} \tag{2.3.6}$$

for some function a.

See [63] for more information on the analytical properties of convex ground costs in optimal transport.

During the past three decades, optimal mass transport has been the subject of intense research. Key works on the subject include two books by Fields medalist Cédric Villani [109, 110], a two-volume set by Svetlozar Rachev and Ludger Rüschendorf [98], and a monograph by Wilfrid Gangbo and Robert J. McCann [63].

Because of their analytical focus, many of these works allow general ground cost functions and assume only that X and Y are Polish spaces. However, a great deal of analytical work has also been done on the L^1 theory, which assumes that the ground cost is a distance. Good overviews on the subject include [1, 48].

2.4 Numerical approaches to the Monge-Kantorovich problem

Numerical work on the Monge-Kantorovich problem predates the modern computer era. Discrete methods were created in the early 20th century, and were one of the motivations for the creation of linear programming. Numerical methods for the continuous Monge-Kantorovich problem have been receiving considerable attention in the last thirty years.

Active research groups have been formed on the theme of numerical optimal transport for continuous and semi-discrete systems, as evidenced by the coordinated effort under the `Mokaplan` umbrella; see [83, 84, 85]. To date, most continuous numerical techniques have been largely restricted to problems with well-behaved cost functions, either the squared or regularized Euclidean distance.

To date, numerical optimal transport research has focused on two specific cases of the cost function:

$$c(\mathbf{x}, \mathbf{y}) = \|\mathbf{y} - \mathbf{x}\|_2^q, \qquad \text{with } q = 2 \text{ or } q = 1. \tag{2.4.1}$$

When $q = 2$, the cost function is strictly convex, and this has a number of important repercussions:

1. the optimal transport plan π^* is unique;
2. the optimal transport plan has an optimal map, T^*;
3. the optimal map T^* is the gradient of some convex function Ψ.

When $q = 1$ in Equation (2.4.1), the cost is not strictly convex, and as a consequence the optimal transport plan π^* may not be unique, and not every optimal transport plan has an optimal map, T^*.

2.4.1 Discrete methods

When transport problems are discrete, they fall into a specific class of problems known as *network flow problems*. As described in [74], there are over 20 established network flow maximization (minimization) techniques, and at least seven publicly-available software packages capable of handling one or more of them. This does not include methods specific to sub-types of transport problems, such as Kuhn's Hungarian Method for the assignment problem [75].

These methods can be generalized into distinct classes. I list them here, along with papers that describe specific instances of each type.

- Successive shortest path algorithms [69]
- Cycle-canceling algorithms [73]
- Capacity-scaling algorithms [46]
- Cost-scaling algorithms [20]
- Cancel-and-tighten algorithms [64]
- Network simplex algorithms [95]

- Primal-dual algorithms [55]
- Out-of-kilter algorithms [59]
- Relaxation algorithms [18]

The first six are summarized in [74], which also contains detailed breakdowns of their worst-case complexity ratings, and the complexity ratings of many other algorithms. The algorithms have different characteristics, depending on the structure of the underlying problem. Given a relatively balanced complete problem, such as those generated by discretizing continuous problems, one can assume the number of nodes is $M + N \sim \mathcal{O}(N)$ and the number of arcs is $MN \sim \mathcal{O}(N^2)$. With this assumption, a typical worst-case complexity is $\mathcal{O}(N^3 \log N)$, given by (for example) [64].

Of course, this is not the complexity of such algorithms in day-to-day applications. As Bertsekas points out, the network simplex algorithm has exponential worst-case behavior, but it is widely used because of its excellent computational properties [15].

2.4.2 Continuous methods

Numerical optimal transport for the continuous problem is a developing area, and to the best of my knowledge there is no concise summary of the existing methods. However, I would group the existing methods into four general categories, listed below with key examples / contributors:

1. Gradient descent methods
    - Bosc [21]
    - Chartrand et al. [35]
    - JKO [70]
2. Augmented Lagrangian methods
    - Haber, Tannenbaum, et al. [66]
    - Benamou [7, 10]
3. Finite difference methods

- BFO [13, 14, 58]
- Regularized Newton's method [79]
- Picard iteration [23]

4. Finite element methods
    - Raviart–Thomas basis functions [5]
5. Projection methods
    - IPFP / Sinkhorn distances [42]
    - Iterative Bregman projections with entropic regularization [11]

Other approaches are in development: e.g., the primal-dual algorithm in [76].

The continuous methods all semi-discretize the problem: namely they discretize $(Y,\, \nu)$, replacing Y by a discrete analog (a mesh) and ν by a finite sum of measures. For example, this can be done by using a quadrature rule on a Cartesian mesh over Y.

Unfortunately, to satisfy well-posedness requirements, these methods must still satisfy the MTW conditions mentioned in Section 2.3. If the ground cost is not strictly convex or c-convex, then c must be regularized in some way. The projection methods mentioned above use entropic regularization, while the Newton's method of Barrett and Prigozhin applies its own custom regularization scheme.

Even when the MTW conditions are satisfied, these methods can only compute over regions where the transport solution must be unique. That requires them to maintain some buffer against region boundaries and areas of μ-measure zero. Otherwise, the associated singularities could distort results.

To date, these restrictions have largely confined the application of continuous methods to well-behaved cost functions such as a squared or regularized Euclidean distance, with carefully constructed domain grids. Continuous approaches focus less on complexity analysis, but typical is the claim of quadratic complexity in [42].

2.5 Applications

Continuous optimal transport is at the center of an interplay between fluid mechanics, partial differential equations, geometry, functional analysis, and probability theory. This creates multiple opportunities for the creation of numerical transport applications.

A substantial amount of image-related research has been done, including work on: image processing [43], image retrieval [100], image warping [35, 113, 114], image registration [3, 67], texture mixing [52, 97], texture synthesis [107], color transfer [51], color image processing [54], and histogram comparison [77].

Medical research is another active field, including: respiratory disease [88, 91], tumor growth modeling [80], structural brain disease [88], electrophysiology of the brain [90, 88], medical imaging [3, 66, 67], and the impact of various clinical treatments [105].

Research in finance and economics includes work being done on transit pricing [26], the principal agent problem [31, 53], and matching problems [33, 61, 62], along with more general works [6, 32, 60]. This research is in addition to interest in game theory: topics like mean-field games [10] and Cournot-Nash equilibria [19].

Computer science [2], robotics [78], and machine learning problems such as risk assessment [108] have generated a lot of interest, as have communications topics: communication networks [87], sensor networks [4], and telecommunications [99]. Physics research is being done on fluid mechanics [8, 9, 30, 37, 96] and quantum mechanics [12, 27, 70].

A range of more general topics have also benefited from transport-related research: civil engineering [89], hydrology [47], general time series data [82, 92], urban planning [26, 34], optics [28, 29], meteorology [41, 40, 39, 68], oceanography [40], sandpiles [50], cosmology [57], and mesh generation [25].

CHAPTER 3
AUCTION ALGORITHMS

The general auction method is best understood by contrasting it with the assignment auction method and its extensions. In order to highlight these contrasts, I will apply every method using a common framework: the general transport problem. This common frame will unify notation and interrelate key concepts. When additional restrictions limit use to a special-case of the transport problem, the required conditions will be described.

3.1 Introduction

The auction method was first proposed by Dimitri Bertsekas in the 1970s. He developed the method for the assignment problem, as an alternative to the Hungarian method [75]. The original auction method has since been extended to solve minimal cost flow and discrete optimal transport problems by taking into account what Bertsekas and Castañón call "similar persons and objects." This extended auction method decomposes the transport problem into an equivalent assignment problem by making copies of each source and sink. (The number of copies is equal to the supply or demand "weight.") It then solves the assignment problem and combines the resulting assignments to provide the desired solution. (The assignment problem generated can be quite large.) All the auction methods offer worst-case error and complexity bounds, as well as conditions under which an optimal solution is guaranteed.

However, auction methods put some restrictions on the data inherent to the problems they can solve. The complexity argument for the assignment auction method requires that the "cost" of each matching is an integer. If some are rational, the costs can be transformed into integers using a common denominator. This impacts the worst-case time complexity of the algorithm, but not its storage requirements. If any cost is irrational, as can occur

when (for instance) they are given by a p-norm with $p \neq 1$, the method offers no formally guaranteed worst-case complexity.

When the extended auction method is used to solve minimal cost flow or optimal transport, then another, more serious, data restriction arises. The extended auction requires that all weights in the network be integral. If any weight is rational, the set of weights can be transformed into integers with a common denominator. Here, though, the common denominator directly impacts both the storage requirements and the worst-case time complexity of the resulting assignment problem. If any weight is irrational, the auction method cannot be applied directly.

3.2 Auction for the assignment problem

In order to understand auction methods for the transport problem, it is necessary to first consider the assignment auction from which they descend. Dimitri Bertsekas proposed the assignment auction method in 1979. Rather than addressing the transport problem, the method was intended to solve the classical assignment problem.

3.2.1 Description and terminology

The assignment problem, as formulated for the original auction method, assumes N sources and N sinks, each of which has weight 1, and integer-valued costs. Thus, the assignment auction method operates on a special case of the discrete transport problem: one with integer costs and unit weights, where $M = N$. Applying the terminology used by Bertsekas, call a sink of weight 1 a *person* and a source of weight 1 an *object*. This semantic distinction, while irrelevant here, will be useful when transitioning to discussing the extended auction for the transport problem.

The auction method solves the dual problem described in Section 2.2.3, using a technique inspired by the real-world process for open ascending price auctions which is commonly used today. We treat each person i as a *bidder* in the auction, seeking to satisfy its

demand for one object, and each object as a *lot* available for purchase. Each unsatisfied bidder i offers a *bid amount*, b_{ij}, for some lot j. The bidders wants to minimize their *loss*; that is, the quantity

$$c_{ij} - b_{ij} \tag{3.2.1}$$

representing the cost-price total for bidder i to obtain lot j. The best possible loss for bidder i is the amount closest to zero, as given by the negative scalar

$$\max_{j \in A(i)} \{c_{ij} - b_{ij}\}. \tag{3.2.2}$$

To get a sense of how the bidding works, consider how real-world bidders make purchasing decisions on online auction sites. The amount bid is given by b_{ij}, while c_{ij} is some shipping and handling charge that will be added to the winning bid. If the shipping cost is high, as it would be for an international purchase, an object may be undesirable even at a very low price, while a low shipping cost might encourage higher-priced bidding. Because the distance between bidder and seller varies, the shipping cost for an object can differ depending on the bidder, influencing how much each person is willing to bid.

Each bid can be "outbid"; that is, superseded by another bidder offering a higher price. So long as b_{ij} is greater than the highest previous price for object j, designated p_j, some bidder (either i or a competitor) will claim lot j. Every object is eventually claimed, regardless of cost, once the prices become sufficiently high. At this point, the auction ends.

Knowing that one requires $b_{ij} > p_j$, the natural question to ask is: how much larger than p_j should one make b_{ij}? As Bertsekas so eloquently explains, one needs to set a minimum *bidding increment*, or *step size*, some $\varepsilon > 0$, such that $b_{ij} \geq p_j + \varepsilon$, or else the auction risks stalling when two options are equally optimal.

One can see by observing real-world auctions that the faster the price is increased, the more quickly the auction is resolved. Thus, it is best to determine an appropriately large bid increment, and to apply that in all calculations. For some bidder i, determine an appropriate

price increase by looking at *opportunity cost*; that is, the difference between the largest and second-largest expense $c_{ij} - p_j$. The desired price increase is equal to this difference plus the bidding increment. Hence, finding out which object to bid on requires knowing the largest net cost, but determining an appropriate price increase requires knowing the second largest net cost.

3.2.2 Iteration

To initialize the method, one must have a bidding step size ε and an initial price vector p. The initial transport map T is assumed to be empty. From then on, the auction is performed in iterations. Each iteration consists of two phases: a bidding phase and a claims phase. The details of each phase are based on [15, pp. 253–254], edited below for consistency of notation.

Bidding phase of the assignment auction: Let $\tilde{I} \subseteq I$ be a nonempty subset of persons i that are unsatisfied under the transport plan T. For each person $i \in \tilde{I}$:

(1) Find:

(a) The object j_i offering best expense, given by

$$j_i = \arg\max_{j \in A(i)} \{c_{ij} - p_j\} \tag{3.2.3}$$

(b) The second-best expense, chosen by considering objects other than j_i,

$$w_i = \max_{j \in A(i),\, j \neq j_i} \{c_{ij} - p_j\} \tag{3.2.4}$$

If j_i is the only object in $A(i)$, define w_i to be $-\infty$. (For computational purposes, this can be any value satisfying $w_i \ll c_{ij_i} - p_{j_i}$.)

(2) Compute the bid $(b_{ij_i},\ q_{ij_i})$, where the bid amount b_{ij_i} is given by

$$b_{ij_i} = c_{ij_i} - w_i + \varepsilon \tag{3.2.5}$$

and the quantity q_{ij_i} is the constant

$$q_{ij_i} = 1. \tag{3.2.6}$$

Claims phase of the assignment auction: For each object j, let $I_j \subseteq I$ be the set of persons from which j received a bid during the bidding phase of the iteration. If I_j is nonempty:

(1) Find:

(a) The person i_j offering the best bid, given by

$$i_j = \underset{\{(b_{ij},\, q_{ij}) \in I_j\}}{\arg\max}\ b_{ij} \tag{3.2.7}$$

(b) The corresponding best bid, given by

$$b_{i_j j} = \max_{\{(b_{ij},\, q_{ij}) \in I_j\}} b_{ij} \tag{3.2.8}$$

(2) Increase p_j to the best bid amount

$$p_j = b_{i_j j}. \tag{3.2.9}$$

(3) If $(k,\ j,\ 1) \in T$:

(a) Add 1 to D_k.

(b) Remove $(k,\ j,\ 1)$ from T.

(4) Append $(i_j,\ j,\ 1)$ to T.

(5) Subtract 1 from D_{i_j}.

3.2.2.1 Noteworthy features

It is readily apparent from the iteration steps above that, for each person i, D_i must be either 0 or 1, depending on whether or not person i has claimed some object j. Given the origins of the auction method, most references will describe person i as *assigned* when $D_i = 1$, and *unassigned* when $D_i = 0$. (The same observation and terminology applies to object j and S_j.) Knowing these terms may be useful when reading other sources, but I will avoid using them because of their potential ambiguity when applied in the non-binary context of the general auction method.

Also consider that the iteration steps do not indicate a definite order in which steps must be performed. In [15], Bertsekas suggests two primary methods of iteration:

1. The *Gauss-Seidel version*, in which the set $\tilde{I}$ consists of a single unsatisfied bidder. Thus, a single iteration looks like:
   (a) Bidding phase: one person bids on a single object.
   (b) Claims phase: that object is claimed by its bidder.
2. The *Jacobi version*, in which the set $\tilde{I}$ consists of a all unsatisfied bidders. Thus, a single iteration is:
   (a) Bidding phase: all persons bid on objects (one object each).
   (b) Claims phase: all objects with bids are claimed by persons.

The iteration steps can be combined in other, more complex, ways. As I will discuss below, they are also naturally suited to parallel and asynchronous implementations.

3.2.3 ε-complementary slackness

One can relax the complementary slackness condition, allowing flow to sources from sinks whenever the loss comes within ε of attaining the maximum. This condition is called *ε-complementary slackness*, or *ε-CS*, and it can be considered for any transport plan, complete or not.

Formally: Given some $\varepsilon > 0$, a simplified transport plan T and price vector p satisfy ε-complementary slackness if

$$x_i - \varepsilon = \max_{k \in A(i)} \{c_{ik} - p_k\} - \varepsilon \leq c_{ij} - p_j \qquad \forall\, (i,\, j,\, f_{ij}) \in T. \tag{3.2.10}$$

3.2.4 Termination and optimality of the assignment auction

The proofs below are all derived, at least in part, from [15], but [15] details some more formally than others. In order to make the underlying dependencies more clear to the mathematical reader, I have elaborated on their arguments in various ways.

3.2.4.1 Assignment auction prices nondecreasing

The proof below, describing essential characteristics of the prices p_j, is my own restatement of an argument given as observation (b) in [15, p. 256].

Theorem 3.2.1. *When applying the assignment auction method, object prices are nondecreasing, and each time an object receives a bid its price increases by at least ε.*

Proof. Assume nonnegative ε is given. Fix the object j^* and let its price before and after iteration be given by p_{j^*} and p'_{j^*}, respectively. If no person bids on j^*, then the price of j^* is unchanged, so $p'_{j^*} = p_{j^*}$. Suppose instead that some person i bids on j^*, and the bid b_{ij^*} is determined to be highest during the claims phase. From Equation (3.2.3), the expense of person i's bid is given by

$$x_i = \max_{j \in A(i)} \{c_{ij} - p_j\} = c_{ij^*} - p_{j^*} \tag{3.2.11}$$

Thus, the bid price b_{ij^*} is given by

$$b_{ij^*} = c_{ij^*} - w_i + \varepsilon = p_{j^*} + x_i - w_i + \varepsilon. \tag{3.2.12}$$

By Equation (3.2.4), $x_i - w_i \geq 0$, so

$$p'_{j^*} = p_{j^*} + x_i - w_i + \varepsilon \geq p_{j^*} + \varepsilon \geq p_{j^*} \tag{3.2.13}$$

Therefore, $p'_{j^*} \geq p_{j^*}$. □

3.2.4.2 Assignment auction expenses nonincreasing

The proof below is my restatement of an argument about the expenses x_i, made as observation (c) in [15, p. 256]. Like Theorem 3.2.1, Theorem 3.2.2 guarantees a worst-case progression of values as the auction iterates.

Theorem 3.2.2. *When applying the assignment auction method, the expense for each person i given by*

$$x_i = \max_{j \in A(i)} \{c_{ij} - p_j\} \tag{3.2.14}$$

is nonincreasing, and every $|A(i)|$ iterations it must decrease by at least ε.

Proof. Given person i, let x_i and x'_i be the expense calculated before and after some iteration, and let p and p' be the price vector before and after that same iteration. By Theorem 3.2.1, for all $j \in A(i)$, $c_{ij} - p_j \geq c_{ij} - p'_j$. Hence,

$$x_i = \max_{j \in A(i)} \{c_{ij} - p_j\} \geq \max_{j \in A(i)} \{c_{ij} - p'_j\} = x'_i. \tag{3.2.15}$$

Thus, the expense x_i is nonincreasing.

Let j_i be the element associated with x_i:

$$x_i = \max_{j \in A(i)} \{c_{ij} - p_j\} = c_{ij_i} - p_{j_i}. \tag{3.2.16}$$

Suppose a bid by person i is accepted for object j_i. Then by Theorem 3.2.1, the price of object j_i increases by at least ε, which implies $c_{ij_i} - p_{j_i}$ decreases by at least ε. Hence, j_i

will not receive another bid from person i until x_i decreases by at least ε. Because there are $|A(i)|$ objects on which person i can bid, this implies that every $|A(i)|$ iterations x_i must decrease by at least ε. □

3.2.4.3 Assignment auction terminates

The following theorem is based on Proposition 7.2 in [15, p. 256–257]. In that text the proof is given as a combination of observations and general statements. I have revised the wording significantly, but the underlying arguments remain the same as in [15].

Theorem 3.2.3. *If at least one feasible transport plan exists, then the assignment auction method terminates.*

Proof. Suppose, by contradiction, that the algorithm does not terminate. Let J^∞ be the subset of objects that received an infinite number of bids, and I^∞ the subset of persons that bid infinitely many times. Since the algorithm is nonterminating, both J^∞ and I^∞ must be nonempty.

As expressed in Theorem 3.2.1, each bid increases an object price by at least ε, so $j \in J^\infty$ implies $p_j \to \infty$. By Theorem 3.2.2, every $|A(i)|$ iterations x_i must decrease by at least ε. Therefore, for all $i \in I^\infty$,

$$x_i = \max_{j \in A(i)} = \{c_{ij} - p_j\} \to -\infty. \tag{3.2.17}$$

Suppose for some person $i \in I^\infty$ there exists an object $j^* \in A(i) \setminus J^\infty$. This implies j^* must be satisfied after a finite number of iterations, so $p_{j^*} < \infty$. It follows that

$$x_i = \max_{k \in A(i)} \{c_{ik} - p_k\} \geq c_{ij^*} - p_{j^*} > -\infty. \tag{3.2.18}$$

This contradicts the assertion that $x_i \to -\infty$, and therefore

$$A(i) \subseteq J^\infty \qquad \forall\, i \in I^\infty. \tag{3.2.19}$$

After a finite number of iterations, each object in J^∞ will be satisfied by a person in I^∞, because the expenses of persons not in I^∞ remain bounded, while the prices of objects in J^∞ increase to $+\infty$.

Furthermore, because the algorithm does not terminate, after a finite number of iterations there must be at least one person in I^∞ that is not satisfied by any object in J^∞, while all persons not in I^∞ have been satisfied. It follows that the number of persons in I^∞ must be strictly larger than the number of objects in J^∞. However, by Equation (3.2.19), persons in I^∞ can only be satisfied by objects in J^∞. This contradicts the assumption that a feasible transport plan exists. Therefore, the algorithm must terminate. □

3.2.4.4 Assignment auction preserves ε-CS

The next argument is taken from [15, p. 255–256], where it appears as Proposition 7.1. It justifies the assumption that the assignment auction preserves the ε-complementary slackness condition, which allows bounding the error on the auction's result. The wording of the argument has been revised slightly and edited for notation.

Theorem 3.2.4. *If a transport plan and price vector satisfy ε-CS for the assignment auction method at the start of an iteration, the same is true of the transport plan and price vector obtained at the end of that iteration.*

Proof. Fix the object j^* and let its price before and after iteration be given by p_{j^*} and p'_{j^*}, respectively. Suppose that person i bids on object j^* during the iteration, and i attains the highest bid. Then by Equations (3.2.4) and (3.2.8)

$$p'_{j^*} = c_{ij^*} - w_i + \varepsilon. \tag{3.2.20}$$

This implies

$$c_{ij^*} - p'_{j^*} = w_i - \varepsilon = \max_{j \in A(i),\, j \neq j^*} \{c_{ij} - p_j\} - \varepsilon. \tag{3.2.21}$$

By Theorem 3.2.1, $p'_j \geq p_j$ for all j, so

$$c_{ij} - p_j \geq c_{ij} - p'_j \qquad \forall\, j \in \mathbb{N}_N. \tag{3.2.22}$$

For any $\varepsilon \geq 0$, it is also the case that

$$c_{ij^*} - p'_{j^*} \geq c_{ij^*} - p'_{j^*} - \varepsilon.$$

Therefore,

$$c_{ij^*} - p'_{j^*} = \max_{j \in A(i),\, j \neq j^*} \{c_{ij} - p_j\} - \varepsilon \geq \max_{j \in A(i),\, j \neq j^*} \{c_{ij} - p'_j\} - \varepsilon \geq \max_{j \in A(i)} \{c_{ij} - p'_j\} - \varepsilon \tag{3.2.23}$$

Suppose that $(i,\, j^*) \in T$ at the start and end of the iteration. Then it must be that no person bid on j^*. This implies the price of j^* is unchanged, so $p'_{j^*} = p_{j^*}$. Because ε-CS held prior to the iteration and $p_j \geq p'_j$ for all j, the ε-CS condition must still hold. Therefore, for all $(i,\, j) \in T$, the ε-CS condition holds after iteration. □

3.2.4.5 Assignment auction error bound

Using ε-CS, the next theorem gives a bound on the error from the assignment auction. The argument is taken from [15, pp. 32–33], where it appears as Proposition 1.4. I have made slight changes to the wording and notation.

Theorem 3.2.5. *If at least one feasible transport plan exists, then when the assignment auction method terminates, the resulting feasible transport plan is within $N\varepsilon$ of optimal.*

Proof. Assume the transport problem is feasible. Let P^* be the optimal primal solution to the assignment problem,

$$P^* = \max_{(i,j) \in \mathcal{A}} c_{ij} f_{ij} = \max_{\substack{j_i,\, i \in \mathbb{N}_N \\ j_i \neq j_k \text{ if } i \neq k}} c_{ij_i}, \tag{3.2.24}$$

and D^* be the optimal dual solution

$$\begin{aligned} D^* &= \min_{p=\{p_j\}_{j=1}^N} \left\{ \sum_{i=1}^{M} d_i \max_{j \in A(i)} \{c_{ij} - p_j\} + \sum_{j=1}^{N} s_j p_j \right\} \\ &= \min_{\substack{p_j \\ j \in \mathbb{N}_N}} \left\{ \sum_{i=1}^{N} \max_{j \in A(i)} \{c_{ij} - p_j\} + \sum_{j=1}^{N} p_j \right\}. \end{aligned} \tag{3.2.25}$$

Suppose $T^* = \{ (i, j_i, 1) \mid i = 1, \ldots, N \}$ is the resulting transport plan when the auction terminates, and $p^* = (p_1^*, \ldots, p_N^*)$ is the resulting price vector. Because (T^*, p^*) satisfies ε-CS,

$$\max_{j \in A(i)} \{c_{ij} - p_j^*\} - \varepsilon \leq c_{ij_i} - p_{j_i}^*. \tag{3.2.26}$$

Therefore,

$$P^* = D^* \leq \sum_{i=1}^{N} \left(\max_{j \in A(i)} \{c_{ij} - p_j^*\} + p_j^* \right) \leq \sum_{i=1}^{N} (c_{ij_i} + \varepsilon) \leq P^* + N\varepsilon = D^* + N\varepsilon. \tag{3.2.27}$$

Therefore, the total transport cost of T^*,

$$\sum_{i=1}^{N} c_{ij_i}, \tag{3.2.28}$$

is within $N\varepsilon$ of the optimal primal cost P^*, and the total transport profit of p^*,

$$\sum_{i=1}^{N} \left(\max_{j \in A(i)} \{c_{ij} - p_j^*\} + p_j^* \right), \tag{3.2.29}$$

is within $N\varepsilon$ of the optimal dual profit D^*. □

3.2.4.6 Assignment auction optimality guarantee

The optimality condition for the assignment auction, given below as Corollary 3.2.6, is based on the first paragraph of [15, p. 34].

Corollary 3.2.6. *If the costs of the assignment problem are all integers, at least one feasible transport plan exists, and $\varepsilon < 1/N$, then when the assignment auction method terminates, the resulting feasible transport plan is optimal.*

Proof. Let P^* be the optimal primal cost of $\mathcal{T}$ and $T^* = \{ (i, j_i, 1) \mid i = 1, \ldots, N \}$ be the transport plan resulting from the auction method applied to $\mathcal{T}$. If $\varepsilon < 1/N$, then $N\varepsilon < 1$, so

$$P^* \leq \sum_{i=1}^{N} c_{ij_i} < P^* + 1. \tag{3.2.30}$$

Because c_{ij} is integral for all $(i, j) \in \mathcal{A}$, it follows that

$$\sum_{i=1}^{N} c_{ij_i} = P^*. \tag{3.2.31}$$

□

3.2.5 Complexity of the auction method for assignment

Understanding the complexity of the assignment auction method requires two additional tools: candidate lists and ε-scaling.

As described in Theorem 3.2.2, expense decreases by at least ε every $|A(i)|$ iterations. By generating an additional structure, called a *candidate list*, $\mathrm{Cand}(i)$, for each bidder i, it is possible to guarantee that expense decreases by at least ε with every iteration.

The practice of ε-scaling consists of applying the auction algorithm several times, with different ε values for each iteration, until the resulting transport plan is optimal. We call each application of the algorithm a *scaling phase*. The price vector resulting from each scaling phase provides good initial prices for the next application.

The complexity argument also requires a number of simplifying assumptions, some very specific, which I detail below.

3.2.5.1 Candidate lists

As was mentioned in Theorem 3.2.2, it is possible that multiple objects j satisfy the expense x_i. Suppose that each time person i scans the objects $j \in A(i)$ to calculate a bid for the best object j_i, one records a list $\mathrm{Cand}(i)$ of pairs $(j,\, p_j^i)$, where

$$j \neq j_i \text{ such that } c_{ij} - p_j = c_{ij_i} - p_{j_i} \tag{3.2.32}$$

and p_j^i is the price of j at the time of the scan by i. This has the potential to be beneficial in two ways: If person i is unassigned and $\mathrm{Cand}(i)$ contains an object j with $p_j = p_j^i$, one knows j must be the best object for i. Furthermore, if there is a second object in $\mathrm{Cand}(i)$, k, with $p_k = p_k^i$, one knows the bidding increment must be exactly ε (since $w_i = x_i$). I will formalize this with the following modifications to the bidding phase.

Bidding phase with candidate lists: For some person i unsatisfied under plan T:

(1) Examine the pairs $(j,\, p_j^i) \in \mathrm{Cand}(i)$ one at a time, until reaching the *second* element for which $p_j^i = p_j$, or until the list's end is reached. If an examined element has $p_j^i < p_j$, discard it.

(2) Branch based on the number of elements found such that $p_j^i = p_j$:

  (a) If two such elements were found:

    (i) Let j_i be the first element on the list for which $p_j^i = p_j$.

    (ii) Discard the contents of the candidate list up to, but not including, the second element.

    (iii) Place a bid on object j_i at price $p_{j_i} + \varepsilon$.

  (b) If fewer than two such elements were found:

    (i) Discard the list $\mathrm{Cand}(i)$.

(ii) Scan the objects in $A(i)$ as described for the "Bidding phase of the assignment auction," determining an object j_i maximizing expense and its corresponding bid $(b_{ij_i},\, q_{ij_i})$.

(iii) Construct a new candidate list consisting of all objects that are tied at expense level w_i (other than j_i) and their current prices.

Because the objects in $A(i)$ are scanned only if $\mathrm{Cand}(i)$ contains fewer than two elements tied for maximum expense, in order for x_i to decrease it is sufficient to scan $A(i)$ once.

3.2.5.2 ε-scaling

Bertsekas found that the number of iterations of the auction algorithm depends strongly on ε and the number of possible values that the cost can take. The latter is called the *cost range*, C, or simply the *range*. When costs are restricted exclusively to the negative (or positive) integers, as they are for Bertsekas, one can generally assume that the cost range C is equal to the maximum absolute cost,

$$C = \max_{(i,j)\in\mathcal{A}} |c_{ij}| = \max_{(i,j)\in\mathcal{A}} \{-c_{ij}\}. \tag{3.2.33}$$

(It is possible for the cost range to be smaller; for example, when $\gcd\{c_{ij}\} > 1$.) As Bertsekas discovered, for many assignment problems the number of iterations required for termination is proportional to C/ε [15, p. 34].

Of course, the number of iterations is also dependent on the initial price vector; when the initial prices are close to "ε-optimal," the number of iterations required is relatively small. This suggests that it may be advantageous to use a scaling technique, similar to that used in penalty and barrier methods. For the auction algorithm, he calls this technique *ε-scaling*.

To perform ε-scaling, apply the auction algorithm multiple times. In the first iteration, use a simple set of initial prices, along with an initial ε. For each successive iteration, use the resulting price vector from the previous application, along with an altered ε-value. The ε-scaling technique generally terminates when ε is reduced below some critical value, such as the $1/N$ desired by the assignment auction method.

Bertsekas suggests the following standard method of ε-scaling (though he suggests that other methods are possible; see [15, p. 260]): Suppose ε^0 is the initial ε value. Assume that ε^k, the value of ε at the $(k+1)$st scaling phase, is given by

$$\varepsilon^k = \frac{\varepsilon^{k-1}}{\theta}, \qquad k = 1,\, 2,\, \ldots,\, K, \tag{3.2.34}$$

where $\theta > 1$ and $K \geq 1$ are some fixed integers. Denote the final value of ε, used in the last scaling phase, as $\overline{\varepsilon}$.

3.2.5.3 Simplifying assumptions

Along with candidate lists and ε-scaling, the complexity argument for the assignment auction method makes a number of simplifying assumptions. These are worth outlining here, as some of them will not apply directly to the general auction method.

(a) The Gauss-Seidel version is implemented; that is, only one person bids during each iteration.

(b) Candidate lists are used to ensure bids are efficiently computed.

(c) Each scaling phase begins with all lots unclaimed.

(d) The initial prices for the first scaling phase are zero, and the initial prices for each subsequent scaling phase are the resulting prices from the previous scaling phase.

(e) For the first scaling phase, the value of ε^0 is a constant fraction of the cost range C.

(f) For the final scaling phase, the value of $\overline{\varepsilon}$ is chosen such that every c_{ij} is an integer multiple of $\overline{\varepsilon}$.

(g) At the beginning of the $(k+1)$st scaling phase, replace each c_{ij} with a corrected value c_{ij}^k that is divisible by ε^k. Corrections should be of size at most ε^k; for example, one may use

$$c_{ij}^k = \left\lceil \frac{c_{ij}}{\varepsilon^k} \right\rceil \varepsilon^k \qquad \forall\,(i,\,j) \in \mathcal{A}, \quad k = 0,\,1,\,\ldots. \tag{3.2.35}$$

Note that correction is unnecessary in the last scaling phase, since Item (f) already requires each c_{ij} be divisible by $\overline{\varepsilon}$.

3.2.5.4 Worst-case complexity of a single scaling phase

The first step in bounding worst-case complexity for the assignment auction method is to consider the complexity of a single scaling phase. The argument is taken from [15, p. 262–263], where it appears as Proposition 7.3. I have changed the notation, and elaborated on some details in order to clarify a few points.

Theorem 3.2.7. *Let $\mathcal{T}$ be a feasible assignment problem having N vertices and $A \leq N^2$ arcs. Suppose one applies the assignment auction method with some $\varepsilon > 0$, and that:*

1. *All the scalars c_{ij} and all the initial object prices are integer multiples of ε;*
2. *For some scalar $r \geq 1$, the initial object prices satisfy $r\varepsilon$-CS together with some feasible assignment.*

Then the running time of the algorithm is $\mathcal{O}(rNA)$.

Proof. Suppose one has transport plan T^0 and price vector p^0 such that $(T^0,\,p^0)$ satisfy $r\varepsilon$-CS. Suppose $(T,\,p)$ is a transport-price pair generated by the auction prior to termination. Thus, T is not complete. For all persons i, define the expense x_i and x_i^0 as

$$x_i = \max_{j \in A(i)} \{c_{ij} - p_j\} \tag{3.2.36}$$

and

$$x_i^0 = \max_{j \in A(i)} \{c_{ij} - p_j^0\}. \tag{3.2.37}$$

By Theorem 3.2.2, all x_i are nonincreasing. I will show that $x_i^0 - x_i$ is bounded above by $(r+1)(N-1)\varepsilon$.

Let i be a person unsatisfied under T. I claim that there exists a path of the form

$$(i,\, j_1,\, i_1,\, \ldots,\, j_n,\, i_n,\, j_{n+1}) \tag{3.2.38}$$

where

1. j_{n+1} is unsatisfied under T;
2. If $n > 0$, then for $k = 1, \ldots, n$, j_k is claimed by i_k under T and j_{k+1} is claimed by i_k under T^0.

This can be shown constructively as follows: Let $k = 0$ and assume $i = i_0$. Then:

(1) Let j_{k+1} be the object claimed by i_k under T^0.

(2) If j_{k+1} is unclaimed under T, stop. Otherwise:

   (a) Increase k by one.

   (b) Let i_k be the person claiming j_k under T. Note that $i_k \neq i_{k-1}$, or one would have stopped in step (2).

   (c) Let j_{k+1} be the object claimed by i_k under T^0. Note that $j_{k+1} \neq j_k$ since j_k is claimed by i_{k-1} under T^0.

   (d) Return to step (2).

This procedure cannot produce the same object twice, so it must terminate, satisfying the properties given, with $0 \leq n \leq N-2$.

Because $(T^0,\, p^0)$ satisfies $r\varepsilon$-CS,

$$x_i^0 = \max_{j \in A(i)} \{c_{ij} - p_j\} \leq c_{ij_1} - p_{j_1}^0 + r\varepsilon \tag{3.2.39}$$

$$c_{i_1 j_1} - p_{j_1}^0 \leq c_{i_1 j_2} - p_{j_2}^0 + r\varepsilon, \tag{3.2.40}$$

$$\cdots \tag{3.2.41}$$

$$c_{i_n j_n} - p_{j_n}^0 \leq c_{i_n j_{n+1}} - p_{j_{n+1}}^0 + r\varepsilon. \tag{3.2.42}$$

Since $(T,\ p)$ satisfies ε-CS,

$$x_i \geq c_{ij_1} - p_{j_1} - \varepsilon \tag{3.2.43}$$

$$c_{i_1 j_1} - p_{j_1} \geq c_{i_1 j_2} - p_{j_2} - \varepsilon, \tag{3.2.44}$$

$$\cdots \tag{3.2.45}$$

$$c_{i_n j_n} - p^0_{j_n} \geq c_{i_n j_{n+1}} - p^0_{j_{n+1}} - \varepsilon. \tag{3.2.46}$$

As shown, j_{n+1} is unassigned under T, so $p_{j_{n+1}} = p^0_{j_{n+1}}$. Thus, by adding the inequalities above,

$$x^0_i - x_i \leq (r+1)(n+1)\varepsilon \leq (r+1)(N-1)\varepsilon \qquad \forall\, i. \tag{3.2.47}$$

Because the c_{ij} and p_j are all integer multiples of ε, all values of $c_{ij} - p_j$, x_i, and w_i will be integer multiples of ε. This implies subsequent values of p_j, x_i, and w_i will also be integer multiples of ε. Recall that the use of candidate lists $\mathrm{Cand}(i)$ scans the objects in $A(i)$ only once to reduce x_i by at least ε. It follows that the number of operations required to place bids for person i is proportional to $(r+1)(N-1)|A(i)|$. Therefore, the running time of the algorithm is proportional to

$$(r+1)(N-1)\sum_{i=1}^{N}|A(i)| = (r+1)(N-1)A, \tag{3.2.48}$$

or $\mathcal{O}(rNA)$, as claimed. □

3.2.5.5 Worst-case complexity of assignment auction method with ε-scaling

Given the result bounding complexity for a single scaling phase, I can now determine worst-case complexity of the assignment auction method as a whole. The following argument is taken from [15, pp. 264–265], with only minor changes.

Theorem 3.2.8. *Let $\mathcal{T}$ be a feasible assignment problem having N vertices and $A \leq N^2$ arcs. Apply the assignment auction method with some $\varepsilon > 0$, and initial prices and costs*

which are integer multiples of ε. Then the worst-case running time of the algorithm is $\mathcal{O}(NA\log(NC))$, where $C = \max_{(i,\,j)}|c_{ij}|$.

Proof. By construction, the c_{ij}^0 and the initial prices in the first scaling phase are integer multiples of ε^0. As shown in Theorem 3.2.7, each new set of prices generated within the scaling phase are integer multiples of ε^0. Thus, the final prices of the scaling phase are integer multiples of ε^0, which implies they are integer multiples of $\varepsilon^1 = \varepsilon^0/\theta$ (since θ is an integer). Therefore, the initial prices (and, by construction, the c_{ij}^1) of the second scaling phase are integer multiples of ε^1. Continuing inductively, the costs and prices remain integer multiples of ε throughout the algorithm.

Thus, by Theorem 3.2.7, the complexity of the $(k+1)$st scaling phase is given by $\mathcal{O}(r^kNA)$, where r^k is such that the initial prices p_j^k of the scaling phase satisfy $r^k\varepsilon^k$-CS together with some feasible transport plan T^k, and with respect to the costs c_{ij}^k.

Let T^k be the resulting assignment of the k-th scaling phase, which must satisfy ε^{k-1}-CS (or $\theta\varepsilon^k$-CS) with respect to the costs c_{ij}^{k-1}. For all $(i,\,j) \in \mathcal{A}$,

$$|c_{ij}^k - c_{ij}^{k-1}| \leq |c_{ij}^k - c_{ij}| + |c_{ij} - c_{ij}^{k-1}| \leq \varepsilon^k - \varepsilon^{k-1} = (1+\theta)\varepsilon^k. \tag{3.2.49}$$

Using the definition of ε-CS, the pair $(T^k,\,p^k)$ must satisfy $(\theta + 2(1+\theta))\varepsilon^k$-CS. Thus, one can use $r^k = \theta + 2(1+\theta)$ in the complexity estimate $\mathcal{O}(r^kNA)$ for the $(k+1)$st scaling phase. Because θ is constant, this implies the running time for all scaling phases except the first is $\mathcal{O}(NA)$.

Because ε^0 is a fixed fraction of the cost range C, the initial (zero) price vector will satisfy $r\varepsilon^0$-CS with any feasible assignment, where r is some fixed constant. Thus, the running time for the first scaling phase is also $\mathcal{O}(NA)$.

Because $\varepsilon^k = \theta\varepsilon^{k+1}$ for all $k = 0,\,1,\,\ldots$, the number of scaling phases is $\mathcal{O}(\log(\varepsilon^0/\overline{\varepsilon}))$. Therefore, the running time of the auction method with ε-scaling is $\mathcal{O}(NA\log(\varepsilon^0/\overline{\varepsilon}))$. If

the c_{ij} are integers, $\overline{\varepsilon} = 1/(N+1)$, and ε^0 is some fixed fraction of C, then $\varepsilon^0/\overline{\varepsilon} = \mathcal{O}(NC)$, and an optimal estimate will be found with worst-case $\mathcal{O}(NA\log(NC))$ computation. □

3.2.6 Considerations for the assignment auction method

3.2.6.1 Relevance of the complexity assumptions

While the requirements given in Section 3.2.5 underly the $\mathcal{O}(NA\log(NC))$ worst-case running time estimate, Bertsekas expresses doubt that they are all necessary for good performance [15, p. 260, 265]. While recommending the use of ε-scaling, he specifically discounts the practicality of candidate lists. For most transport problems, ties in the expense x_i occur so infrequently that the overhead involved in maintaining candidate lists far exceeds any savings.

3.2.6.2 Rational values in the assignment auction method

As mentioned by Bertsekas in [15, p. 34], assignment problems with rational costs can be solved using the auction method by employing a scaling process. Multiplying all rational costs by a common denominator in order to get integer values preserves the inherent cost relationships. Thus, any optimal transport plan for the cost-multiplied problem is an optimal transport plan for the original problem, and the optimal primal cost for the original problem can be found by dividing the solution by the common denominator.

3.2.6.3 Irrational values in the assignment auction method

The simplifying assumption that all costs c_{ij} be integer multiples of ε is particularly relevant, because it implicitly restricts the validity of the worst-case complexity argument to rationally-related costs. Bertsekas suggests that the approach of modifying c_{ij} values to make them integer multiples of ε is of questionable practical use.

He also states (but does not prove) that if the c_{ij} are approximated arbitrarily closely by rational numbers, such that all are within $\delta > 0$ of their original values, then the final assignment will be within $N(\varepsilon + \delta)$ of optimality [15, p. 264].

3.3 Extended auction for the transport problem

The extended auction for the integer-valued transport problem was initially described by Bertsekas and Castañón in 1989. They wanted to extend the assignment auction method to one that could handle transport problems. Their idea was to convert the integer-valued transport problem into an assignment problem by creating multiple copies of persons (or objects) for each sink (or source, respectively), and then to modify the method to take advantage of the presence of the multiple copies. After solving the assignment problem, redundant arcs with positive flow are combined to generate the optimal transport plan.

The extended auction requires both costs and weights to be integers. Also, recall the assumption that $M \geq N$; that is, there are at least as many sinks as sources. These two features are relevant to understanding the power and limitations of the method.

3.3.1 Description and terminology

The auction method was originally designed for the assignment problem, so the integer transport problem must first be transformed into an assignment problem. Construct the assignment problem as follows:

(1) For each source j in the transport problem $\mathcal{T}$:

   (a) Put s_j identical objects, $j_1, j_2, \ldots, j_{s_j}$, in the assignment problem $\mathcal{T}^*$, each with associated supply 1.

(2) For each sink i in the transport problem $\mathcal{T}$:

   (a) Put d_i identical persons, $i_1, i_2, \ldots, i_{d_i}$, in the assignment problem $\mathcal{T}^*$, each with associated demand 1.

   (b) For all $k \in \mathbb{N}_{d_i}$, if $j \in A(i)$ then put $j_l \in A^*(i_k)$ for all $l \in \mathbb{N}_{s_j}$.

(3) For each $(i,\ j) \in \mathcal{A}$:

(a) Set $c^*_{i_k j_l} = c_{ij}$ for all $k \in \mathbb{N}_{d_i}$ and $l \in \mathbb{N}_{s_j}$.

Because the integer transport problem had weights satisfying

$$\sum_{i=1}^{M} d_i = \sum_{j=1}^{N} s_j = L > 0, \tag{3.3.1}$$

the assignment problem $\mathcal{T}^*$ has L persons and L objects, and is feasible if $\mathcal{T}$ is feasible.

3.3.1.1 Similar objects and persons

By virtue of the construction described above, the persons / objects of the assignment problem $\mathcal{T}^*$ have a great deal in common. One can formally describe the relationships between persons (and objects) in terms of equivalence classes. Say that two sinks i and i' are *similar*, writing $i \sim i'$, if

$$A(i) = A(i') \tag{3.3.2}$$

$$\text{and} \qquad c_{ij} = c_{i'j} \qquad \text{for all } j \in A(i). \tag{3.3.3}$$

Two sources j and j' are *similar*, denoted $j \sim j'$, if

$$\{\, i \mid j \in A(i) \,\} = \{\, i \mid j' \in A(i) \,\} \tag{3.3.4}$$

$$\text{and} \qquad c_{ij} = c_{ij'} \qquad \text{for all } i \text{ with } j \in A(i). \tag{3.3.5}$$

The extended auction method makes extensive use of these equivalence classes, called *similarity classes*. A similarity class of persons or objects is denoted $S(i)$ or $S(j)$, respectively.

By a slight abuse of notation, one may also use $S(i)$ for some sink i in $\mathcal{T}$ to refer to the class of similar persons in $\mathcal{T}^*$ which were derived from i. In this case, one may mention the similarity class of the sink i, though actually referencing the persons in $\mathcal{T}^*$ derived from a

sink in $\mathcal{T}$. (The same conventions may be used with regard to $S(j)$, where j is some source in $\mathcal{T}$.)

3.3.1.2 ε-CSS

Recall that the auction method solves the dual problem, generating an optimal price vector. The assignment problem $\mathcal{T}^*$ generates price vectors containing L prices, but the dual for the associated transport problem $\mathcal{T}$ requires a price vector containing only N prices. This discrepancy motivates Bertsekas and Castañón to define a new concept, which they call "ε-complementary slackness strengthened." I describe the concepts underlying "ε-CSS" in my own words, below.

Given a price vector p associated with the assignment problem $\mathcal{T}^*$, define the *price of the similarity class* $S(j)$ of an object j as

$$\overline{p}_j = \min_{k \in S(j)} p_k \qquad j = 1, \ldots, L. \tag{3.3.6}$$

One may also refer to the price of the similarity class $S(j)$ as a *similarity price*, and to the vector of such prices as a *similarity price vector*. As is evident, all objects in the same similarity class share the same $\overline{p}_j$. Even though the price vector of object similarity classes has L members, one can easily pick out the N distinct members by considering a single representative from each similarity class.

For any person i associated with $\mathcal{T}^*$, use Equation (2.2.11) to write the expense of i as

$$x_i = \max_{j \in A(i)} \{c_{ij} - p_j\} = \max_{j \in A(i)} \{c_{ij} - \overline{p}_j\}. \tag{3.3.7}$$

It is worth noting that all persons in the same similarity class have the same expense, and that expense x_i is determined by the similarity price. These observations suggest that optimality conditions for the assignment problem will depend solely on the similarity price

vector, and not on the price vector as a whole. To formally address this idea, define a stronger form of complementary slackness.

Given $\varepsilon \geq 0$, a simplified transport plan T and price vector p satisfy *ε-complementary slackness strengthened* (*ε-CSS*) if for all $(i,\ j,\ f_{ij}) \in T$,

$$\begin{cases} c_{ij} - p_j \geq \max_{k \in A(i)} \{c_{ik} - p_k\} - \varepsilon & A(i) \setminus S(j) = \varnothing \\ c_{ij} - p_j \geq \max_{k \in A(i) \setminus S(j)} \{c_{ik} - p_k\} - \varepsilon & \text{otherwise.} \end{cases} \tag{3.3.8}$$

3.3.1.3 ε-CSS implies ε-CS

In most of the algorithms for the extended auction, ε-CS of the pair $(T,\ \overline{p})$ is maintained, but ε-CS may be violated by the pair $(T,\ p)$. Hence, it is worth formalizing the relationship between ε-CSS and ε-CS. As explained in [16, pp. 77–78], ε-CSS implies ε-CS, though the converse is not true. The theorem below restates the argument of [16] in more detail.

Theorem 3.3.1. *If a price vector p and associated transport plan T satisfy ε-CSS, then the pair consisting of T and the similarity price vector $\overline{p}$ satisfies ε-CS.*

Proof. Assume without loss of generality that T is a simplified transport plan, which along with the price vector p satisfies ε-CSS. For all $(i,\ j,\ f_{ij}) \in T$, if $A(i) \setminus S(j) = \varnothing$, the definition of ε-CSS is identical to that of ε-CS. If $A(i) \setminus S(j) \neq \varnothing$, by the definition of ε-CSS and the prices for similarity classes,

$$c_{ij} - \overline{p}_j \geq c_{ij} - p_j \geq \max_{k \in A(i) \setminus S(j)} \{c_{ik} - p_k\} - \varepsilon. \tag{3.3.9}$$

By the definition of $\overline{p}_j$ as a minimum,

$$c_{ij} - \overline{p}_j = \max_{k \in S(j)} \{c_{ik} - p_k\} \geq \max_{k \in S(j)} \{c_{ik} - p_k\} - \varepsilon. \tag{3.3.10}$$

Therefore, for all $(i,\ j,\ f_{ij}) \in T$,

$$c_{ij} - \overline{p}_j \geq \max_{k \in A(j)} \{c_{ik} - p_k\} - \varepsilon = \max_{k \in A(j)} \{c_{ik} - \overline{p}_k\} - \varepsilon, \tag{3.3.11}$$

and so $(T,\ \overline{p})$ satisfies ε-CS. □

3.3.2 Summary of algorithms

As Bertsekas and Castañón themselves recognize [16, p. 72], "the performance of the [assignment auction] method can be quite poor." They address this issue by appealing to the special structure of the transformed transport problem. When sinks and sources are replaced with multiple copies, the resulting persons and objects retain the cost relationships of the originals. Because of these relationships, such persons and objects tend to become involved in protracted "bidding wars" unless that underlying structure is taken into account.

Bertsekas and Castañón address that structure in multiple ways. Thus, the extended auction method is best understood as three distinct algorithms:

- The *auction algorithm for the assignment problem*, or **assignment auction**, expands the sinks and sources of the transport problem, creating an assignment problem that it solves without considering any impact from the underlying structure.
- The *auction algorithm for similar objects*, or **SO auction** considers the impact of similar objects when increasing prices.
- The *auction algorithm for similar objects and persons*, or **SOP auction**, treats similar persons as a unit during each bidding phase. At every iteration, each unsatisfied similarity class of persons bids collectively for a number of objects equal to its total demand d_i. Each similarity class of persons shares a single price increase, determined similarly to the technique of the SO auction.

3.3.3 Iteration

Regardless of the algorithm used, to initialize the extended auction method one must have a bidding step size ε and an initial price vector. From then on, the auction is performed in iterations. Each iteration consists of two phases: a bidding phase and a claims phase. The assignment auction algorithm works exactly as described in Section 3.2.2. The other two algorithms offered for the extended auction are described below.

3.3.3.1 SO auction

Bidding phase of the extended (SO) auction: Let $\tilde{I}$ be a nonempty subset of persons i that are unsatisfied under the transport plan T. For each person $i \in \tilde{I}$:

(1) Find:

(a) The object j_i offering best expense, given by

$$j_i = \arg\max_{j \in A(i)} \{c_{ij} - p_j\} \tag{3.3.12}$$

(b) The second-best expense, chosen by considering objects other than those in the similarity class of j_i,

$$w_i = \max_{j \in A(i) \setminus S(j_i)} \{c_{ij} - p_j\} \tag{3.3.13}$$

If $A(i) \setminus S(j_i) = \varnothing$, define w_i to be $-\infty$. (For computational purposes, this can be any value satisfying $w_i \ll c_{ij_i} - p_{j_i}$.)

(2) Compute the bid $(b_{ij_i},\ q_{ij_i})$, where b_{ij_i} is given by

$$b_{ij_i} = c_{ij_i} - w_i + \varepsilon \tag{3.3.14}$$

and q_{ij_i} is the constant

$$q_{ij_i} = 1. \tag{3.3.15}$$

Claims phase of the extended (SO) auction: For each object j, let I_j be the set of persons from which j received a bid in the bidding phase of the iteration. If I_j is nonempty:

(1) Find:

(a) The person i_j offering the highest bid, given by

$$i_j = \underset{\{(b_{ij}, q_{ij}) \in I_j\}}{\arg\max} \; b_{ij} \tag{3.3.16}$$

(b) The corresponding highest bid, given by

$$b_{i_j j} = \max_{\{(b_{ij}, q_{ij}) \in I_j\}} b_{ij} \tag{3.3.17}$$

(2) Increase p_j to the highest bid

$$p_j = b_{i_j j}. \tag{3.3.18}$$

(3) If $(k,\, j,\, 1) \in T$:

(a) Add 1 to D_k.

(b) Remove $(k,\, j,\, 1)$ from T.

(4) Append $(i_j,\, j,\, 1)$ to T.

(5) Subtract 1 from D_{i_j}.

3.3.3.2 SOP auction

Bidding phase of the extended (SOP) auction: Let $\tilde{I}$ be a nonempty subset of similarity classes of persons $S(i)$, such that each $S(i) \in \tilde{I}$ contains one or more persons that are unsatisfied under the transport plan T. For each similarity class of persons $S(i)$ in $\tilde{I}$ represented by element i:

(1) Let $i_1,\, i_2,\, \ldots,\, i_k$ be the persons in $S(i)$ that are satisfied under T, and $j_1,\, j_2,\, \ldots,\, j_k$ the objects which they have claimed.

Let $i_{k+1},\, \ldots,\, i_l$ be the persons in $S(i)$ that are unsatisfied under T.

Let $j_{k+1}, \ldots, j_{l'}$ be the objects in $A(i)$ that are not claimed by persons in $S(i)$.

For each source $j \in \{j_{k+1}, \ldots, j_{l'}\}$, compute the current expense

$$x_{ij} = c_{ij} - p_j. \tag{3.3.19}$$

(2) Order the expenses

$$x_{ij_{k+1}} \geq x_{ij_{k+2}} \geq \cdots \geq x_{ij_{l'}}. \tag{3.3.20}$$

(3) Compute the scalar w_i as follows:

(a) If $l < l'$ and $j_1, \ldots, j_l$ do not belong to the same similarity class, let

$$w_i = x_{ij_{l+1}}. \tag{3.3.21}$$

(b) If $l < l'$ and $j_1, \ldots, j_l$ belong to the same similarity class, let w_i be the expense x_{ij} of the first object $j \in \{j_{l+1}, \ldots, j_{l'}\}$ that does not belong to the common similarity class $S(j_1)$. If no such object exists, define w_i as given in Equation (3.3.21).

(c) If $l = l'$, define $w_i = -\infty$. For computational purposes, this need only satisfy

$$w_i \ll x_{ij_{l'}}. \tag{3.3.22}$$

(4) For each $r \in \mathbb{N}_l$, compute the bid of person i_r for the object j_r as $(b_{i_r j_r}, q_{i_r j_r})$, where $b_{i_r j_r}$ is given by

$$b_{i_r j_r} = c_{i_r j_r} - w_i + \varepsilon \tag{3.3.23}$$

and $q_{i_r j_r}$ is the constant

$$q_{i_r j_r} = 1. \tag{3.3.24}$$

Claims phase of the extended (SOP) auction: For each object j, let I_j be the set of persons from which j received a bid in the bidding phase of the iteration. If I_j is nonempty:

(1) Find:

(a) The person i_j offering the highest bid, given by

$$i_j = \underset{\{(b_{ij}, q_{ij}) \in I_j\}}{\arg\max} \; b_{ij} \tag{3.3.25}$$

(b) The corresponding highest bid, given by

$$b_{i_j j} = \max_{\{(b_{ij}, q_{ij}) \in I_j\}} b_{ij} \tag{3.3.26}$$

(2) Increase p_j to the highest bid

$$p_j = b_{i_j j}. \tag{3.3.27}$$

(3) If $(k,\, j,\, 1) \in T$:

(a) Add 1 to D_k.

(b) Remove $(k,\, j,\, 1)$ from T.

(4) Append $(i_j,\, j,\, 1)$ to T.

(5) Subtract 1 from D_{i_j}.

3.3.4 Termination and optimality of the SOP auction

The results below are all stated or implied in [16], but some are expressed more formally than others. As written, all of them apply specifically to the SOP auction algorithm; see Section 3.3.6 for details on how the SOP auction relates to the other algorithms. All of the original arguments have been extended or restated to some degree, as explained in detail below.

3.3.4.1 SOP auction prices are nondecreasing

The following result is given as the first part of Proposition 2 in [16, p. 80–82]. The use of similarity classes makes the argument somewhat different than that given in Theorem 3.2.1

(that result may be seen as a special case of Theorem 3.3.2). For this reason, I restate the argument in its entirety. I have included additional details (such as case 1(c)), and made minor organizational changes in order to improve clarity. Notation also differs slightly from the original.

Theorem 3.3.2. *When applying the SOP auction algorithm to a feasible integer transport problem, object prices are nondecreasing, and during each iteration at least one object price increases by at least ε.*

Proof. Suppose ε-CSS holds before some arbitrary iteration, and one has object prices p and p' before and after iteration, respectively. Let T be the transport plan at the start of the iteration.

Suppose person $i_k \in S(i)$ bids for person $j_k \in S(j)$ during the iteration. Either $(i_k, j_k) \in T$ or not.

1. Suppose $(i_k, j_k) \in T$, and assume without loss of generality that $A(i) \setminus S(j) \neq \varnothing$. By ε-CSS

$$c_{i_k j_k} - p_{j_k} \geq \max_{r \in A(i) \setminus S(j)} \{c_{ir} - p_r\} - \varepsilon. \tag{3.3.28}$$

   There are three possibilities for w_i:

   (a) If $l < l'$ and $j_1, \ldots, j_l$ do not belong to the same similarity class,

$$w_i = x_{i j_{l+1}} \leq \max_{r \in A(i) \setminus S(j)} \{c_{ir} - p_r\}. \tag{3.3.29}$$

   (b) If $l < l'$ and $j_1, \ldots, j_l$ belong to the same similarity class,

$$w_i = \max_{r \in A(i) \setminus S(j)} \{c_{ir} - p_r\}. \tag{3.3.30}$$

   (c) If $l = l'$,

$$w_i = -\infty. \tag{3.3.31}$$

In all three cases, by applying Equation (3.3.28) one has

$$c_{i_k j_k} - p_{j_k} \geq w_i - \varepsilon. \tag{3.3.32}$$

Thus, the bid price computed is

$$b_{i_k j_k} = c_{i_k j_k} - w_i + \varepsilon \geq p_{j_k}. \tag{3.3.33}$$

2. If $(i_k,\ j_k) \notin T$, then because of the ordering of expenses given in Equation (3.3.20), one has

$$c_{i_k j_k} - p_{j_k} \geq w_i \tag{3.3.34}$$

for all three possible situations. Therefore,

$$b_{i_k j_k} = c_{i_k j_k} - w_i + \varepsilon \geq p_{j_k} + \varepsilon. \tag{3.3.35}$$

Because the price of j_k is equal to the highest bid, by the two cases above one has

$$p'_{j_k} \geq \begin{cases} p_{j_k} & \text{if } (i_k,\ j_k) \in T \\ p_{j_k} + \varepsilon & \text{if } (i_k,\ j_k) \notin T \end{cases} \tag{3.3.36}$$

If some object j does not receive a bid during the iteration, $p'_j = p_j$. Hence, all object prices are nondecreasing. Furthermore, since at least one unassigned person bids during each iteration, at least one object price must increase by ε. □

3.3.4.2 SOP auction expenses are nonincreasing

The following result is stated as observation (c) in [16, p. 83]. I have written the argument more formally. Because of the use of similarity classes, the details are somewhat

different than that stated in Theorem 3.2.2 (that result may be seen as a special case of Theorem 3.3.3). For this reason, the argument is given in its entirety.

Theorem 3.3.3. *When applying the SOP auction algorithm, the expense for each person i given by*

$$x_i = \max_{j \in A(i)} \{c_{ij} - p_j\} \tag{3.3.37}$$

is nonincreasing, and every $\sum_{j \in A(i)} |S(j)|$ iterations it must decrease by at least ε.

Proof. Given person i, let x_i and x_i' be the expense calculated before and after some iteration, and let p and p' be the price vector before and after that same iteration. By Theorem 3.3.2, for all $j \in A(i)$, $c_{ij} - p_j \geq c_{ij} - p_j'$. Hence,

$$x_i = \max_{j \in A(i)} \{c_{ij} - p_j\} \geq \max_{j \in A(i)} \{c_{ij} - p_j'\} = x_i'. \tag{3.3.38}$$

Thus, the expense x_i is nonincreasing.

Let j_i be the element associated with x_i:

$$x_i = \max_{j \in A(i)} \{c_{ij} - p_j\} = c_{ij_i} - p_{j_i}. \tag{3.3.39}$$

Suppose a bid by person i is accepted for object j_i. Then by Theorem 3.3.2, the price of object j_i increases by at least ε, which implies $c_{ij_i} - p_{j_i}$ decreases by at least ε. Hence, j_i will not receive another bid from person i until x_i decreases by at least ε. Because there are $\sum_{j \in A(i)} |S(j)|$ objects on which person i can bid, this implies that every $\sum_{j \in A(i)} |S(j)|$ iterations x_i must decrease by at least ε. □

3.3.4.3 SOP auction terminates

The SOP termination result is given as Proposition 3 in [16, p. 83]. It is nearly identical to the argument made in Theorem 3.2.3.

Theorem 3.3.4. *If at least one feasible plan exists for the integer transport problem, then the SOP auction algorithm terminates.*

Proof. Suppose, by contradiction, that the algorithm does not terminate. Let J^∞ be the subset of objects that received an infinite number of bids, and I^∞ the subset of persons that bid an infinite number of times. Since the algorithm is nonterminating, both J^∞ and I^∞ must be nonempty.

As expressed in Theorem 3.3.2, each bid increases an object price by at least ε, so $j \in J^\infty$ implies $p_j \to \infty$.

By Theorem 3.3.3, every $\sum_{j \in A(i)} |S(j)|$ iterations x_i must decrease by at least ε. Therefore, for all $i \in I^\infty$,

$$x_i = \max_{j \in A(i)} = \{c_{ij} - p_j\} \to -\infty. \tag{3.3.40}$$

By the definition of the expense as a maximum,

$$A(i) \subseteq J^\infty \qquad \forall\, i \in I^\infty. \tag{3.3.41}$$

Otherwise, x_i would be bounded for some $i \in I^\infty$, a contradiction.

After a finite number of iterations, each object in J^∞ will be satisfied by a person in I^∞, because the expenses of persons not in I^∞ remain bounded, while the prices of objects in J^∞ increase to $+\infty$.

Furthermore, because the algorithm does not terminate, after a finite number of iterations there must be at least one person in I^∞ that is not satisfied by any object in J^∞, while all persons not in I^∞ have been satisfied. It follows that the number of persons in I^∞ must be strictly larger than the number of objects in J^∞. However, by Equation (3.3.41), persons in I^∞ can only be satisfied by objects in J^∞. This contradicts the assumption that a feasible transport plan exists. Therefore, the algorithm must terminate. □

3.3.4.4 SOP auction preserves ε-CSS

The argument that the SOP auction preserves ε-CSS is given as the second part of Proposition 2 in [16, pp. 80, 82]. I have changed the notation for the sake of consistency.

Theorem 3.3.5. *If a transport plan and price vector satisfy ε-CSS for the SOP auction algorithm at the start of an iteration, the same is true of the transport plan and price vector obtained at the end of that iteration.*

Proof. Assume ε-CSS holds before some arbitrary iteration, and one has object prices p and p' before and after iteration, respectively.

Suppose (i^*, j^*) belongs to the transport plan following the iteration, and that i^* bid for j^* during the iteration. If $w_{i^*} = -\infty$ or $A(i^*) \setminus S(j^*) = \varnothing$, then ε-CSS is trivially preserved. Assume instead that $w_{i^*} > -\infty$ and $A(i^*) \setminus S(j^*) \neq \varnothing$.

For all $j \in A(i^*)$ that received a bid from some $i \in S(i^*)$,

$$c_{i^*j^*} - p'_{j^*} = c_{ij} - b_{ij} \geq c_{ij} - p'_j \geq c_{ij} - p'_j - \varepsilon. \tag{3.3.42}$$

Therefore,

$$c_{i^*j^*} - p'_{j^*} \geq c_{i^*j} - p'_j. \tag{3.3.43}$$

Suppose there exists some bidder $k \neq i^*$ and some $j \in A(k) \setminus S(j^*)$ which did not receive a bid from any person in $S(i^*)$. Because $w_{i^*} > -\infty$ and prices are nondecreasing, it must be the case that

$$w_{i^*} \geq c_{i^*j} - p_j \geq c_{i^*j} - p'_j \tag{3.3.44}$$

Then

$$c_{i^*j^*} - p'_{j^*} = c_{i^*j^*} - b_{i^*j^*} = w_{i^*} - \varepsilon \geq c_{i^*j} - p'_j - \varepsilon. \tag{3.3.45}$$

Combining these two possibilities,

$$c_{i^*j^*} - p'_{j^*} \geq \max_{j \in A(i^*) \setminus S(j^*)} \{c_{i^*j} - p'_j\} - \varepsilon. \tag{3.3.46}$$

So ε-CSS is satisfied.

Suppose now that one has $(i^*,\ j^*)$ in the transport plan at the end of the iteration, but that i^* did not bid for j^* during this iteration. This implies all persons in $S(i^*)$ were satisfied during the iteration. Let p^{i^*} be the price vector at the end of the last iteration in which persons in $S(i^*)$ bid. By ε-CSS,

$$c_{i^*j^*} - p^{i^*}_{j^*} \max_{j \in A(i^*) \setminus S(j^*)} \{c_{i^*j} - p^{i^*}_j\} - \varepsilon. \tag{3.3.47}$$

The price of j^* has necessarily remained unchanged since that bidding phase, so $p_{j^*} = p^{i^*}_{j^*}$. Since prices are nondecreasing,

$$p'_j \geq p^{i^*}_j \qquad \forall\, j \in A(i^*) \setminus \{j^*\}. \tag{3.3.48}$$

Therefore, by combining these relations,

$$c_{i^*j^*} - p'_{j^*} \max_{j \in A(i^*) \setminus S(j^*)} \{c_{i^*j} - p'_j\} - \varepsilon. \tag{3.3.49}$$

Thus, ε-CSS holds for all $(i^*,\ j^*)$ in the transport plan at the end of the iteration. □

3.3.4.5 SOP auction optimality guarantee

The optimality result for the SOP auction is given as Proposition 4 in [16, pp. 85–86]. I re-state it here with only minor changes. The argument takes a different form than many of the proceeding auction results, because it relies primarily on concepts used when implementing the network simplex method for the transport problem.

Theorem 3.3.6. *If the costs of the assignment problem are all integers, at least one feasible transport plan exists, and* $\varepsilon < 1/\min\{M, N\}$*, then when the SOP auction algorithm terminates, the resulting feasible transport plan is optimal.*

Proof. To prove this, I draw upon the underlying network structure of the transport graph. Suppose T is not optimal. Then there must exist a simple cycle

$$Y = (i_1, j_2, i_2, j_3, \ldots, i_{k-1}, j_k, i_k, j_1, i_1) \tag{3.3.50}$$

along which flow can be pushed such that the resulting transport plan T' is feasible and the primal cost is increased. Because Y is a simple cycle, it has no repeated vertices, and therefore $k \leq \min\{M, N\}$. The vertices i_m and j_m are sinks and sources, respectively, with $m \in \mathbb{N}_k$. For $m = 1, \ldots, k-1$, one has $j_m \in A(i_m)$ and $j_{m+1} \in A(i_m)$, along with $j_k \in A(i_k)$ and $j_1 \in A(i_k)$.

Because $\varepsilon < 1/\min\{M, N\}$, $k\varepsilon < 1$. Since flow can be pushed from i_m to j_m for all $m \in \mathbb{N}_k$, and all flows must be integral, then

$$f_{i_m j_m} \geq 1 \qquad \forall\, m \in \mathbb{N}_k. \tag{3.3.51}$$

Furthermore, because pushing flow along Y must improve the cost, it must be that

$$\sum_{m=1}^{k} c_{i_m j_m} + 1 \leq c_{i_k j_1} + \sum_{m=2}^{k} c_{i_{m-1} j_m}. \tag{3.3.52}$$

Because the auction method has completed, it follows that the resulting price vector satisfies

$$\sum_{m=1}^{k} (c_{i_m j_m} - p_{j_m}) + 1 \leq (c_{i_k j_1} - p_{j_1}) + \sum_{m=2}^{k} (c_{i_{m-1} j_m} - p_{j_m}) \leq \sum_{m=1}^{k} x_{i_m}. \tag{3.3.53}$$

Since the flow is positive on $f_{i_m j_m}$ for all m, the ε-CS condition must be satisfied for those pairs; that is,

$$x_{i_m} - \varepsilon \leq c_{i_m j_m} - p_{j_m} \qquad \forall\, m \in \mathbb{N}_k. \tag{3.3.54}$$

Therefore,

$$\sum_{m=1}^{k}(c_{i_m j_m} - p_{j_m}) + 1 \leq \sum_{m=1}^{k} x_{i_m} \leq \sum_{m=1}^{k}(c_{i_m j_m} - p_{j_m}) + k\varepsilon < \sum_{m=1}^{k}(c_{i_m j_m} - p_{j_m}) + 1. \tag{3.3.55}$$

This is a contradiction; therefore, T must be optimal. □

3.3.5 Complexity of the SOP auction

Bertsekas and Castañón state in [16, p. 91] that "it is possible to use the algorithm of the previous section to construct an $\mathcal{O}((M + N)^3 \log(C \min\{M, N\}))$ transportation algorithm." Because the proof of this claim, appearing in [17], is given as a corollary to a result for the minimum cost network flow problem. A detailed description of the proof would be a significant digression from the information needed for the optimal transport problem, and so it is not included here.

3.3.6 Relationship of the three extended auction algorithms

Because the extended auction method proposed by Bertsekas and Castañón in [16] offers the reader three distinct algorithms, it is natural to consider how those algorithms are related to one another.

3.3.6.1 Assignment auction is a special case of SO auction

Bertsekas and Castañón describe the SO auction as "a variation" of the assignment auction, but do not elaborate further [16, p. 74]. I offer the following result formalizing the relationship between the two algorithms.

Theorem 3.3.7. *If $S(j) = \{j\}$ for all objects j in $\mathcal{T}$, then the assignment auction algorithm is equivalent to the auction algorithm for similar objects.*

Proof. Let $\mathcal{T}$ be any feasible transport problem such that for all objects j, $S(j) = \{j\}$. Note that the steps of the assignment auction algorithm differ from those of the SO auction only in the definition of w_i.

Let i be any person unsatisfied under the transport plan T, with object j_i offering best expense, and consider the value of w_i. Because $\mathcal{T}$ is feasible, $A(i)$ is nonempty. Since $S(j_i) = \{j_i\}$, this implies $A(i) = \{j_i\}$ if and only if $A(i) \setminus S(j_i) = \varnothing$. Therefore, for both algorithms, the conditions under which $w_i = -\infty$ are equivalent.

Suppose instead that $A(i) \neq \{j_i\}$, which occurs if and only if $A(i) \setminus S(j_i)$ is nonempty. Then

$$w_i = \max_{j \in A(i) \setminus S(j_i)} \{c_{ij} - p_j\} = \max_{j \in A(i),\, j \neq j_i} \{c_{ij} - p_j\} \tag{3.3.56}$$

Therefore, the assignment auction is a special case of the SO auction, and the two are equivalent when $S(j) = \{j\}$ for all objects j. □

3.3.6.2 SO auction is a special case of SOP auction

The relationship between the SO and SOP auction algorithms is stated without proof in [16, p. 80]. I offer a proof for this relationship, as well.

Theorem 3.3.8. *If $S(i) = \{i\}$ for all i in $\mathcal{T}$, then the auction algorithm for similar objects is equivalent to the auction algorithm for similar persons and objects.*

Proof. Let $\mathcal{T}$ be any feasible transport problem such that for all persons i, $S(i) = \{i\}$. Note that the steps of the SO auction algorithm differ from those of the SOP auction only in the bidding phase.

Let $\tilde{I}$ be a nonempty subset of similarity class of persons $S(i)$,

$$\tilde{I} = \{S(i_1),\, S(i_2),\, \ldots,\, S(i_t)\}, \tag{3.3.57}$$

such that for $s = 1, \ldots, t$, $S(i_s) \in \tilde{I}$ contains one or more persons that are unsatisfied under the transport plan T. Let $\tilde{I}'$ be the set

$$\tilde{I}' = \{i_1, i_2, \ldots, i_t\} \tag{3.3.58}$$

For $s = 1, \ldots, t$, $S(i_s) = \{i_s\}$, so $i_s \in \tilde{I}'$ if and only if $S(i_s) \in \tilde{I}$.

Suppose one is bidding for element $S(i)$ in $\tilde{I}$. Since $S(i) = \{i\}$, the definition of $\tilde{I}$ implies that no element in $S(i)$ is satisfied. Thus, $k = 0$, $l = 1$, and $l' = |A(i)|$. This implies $l = l'$ if and only if $|A(i)| = 1$. In this case, the object offering best expense for i in the SO auction must be $A(i) = j_i$, so $A(i) \setminus S(j_i) = \varnothing$ if and only if $l = l'$. Therefore, the conditions under which $w_i = -\infty$ are identical.

Suppose instead that $l < l'$. Because $l = 1$, it must be the case that $j_1, \ldots, j_l$ belong to the same similarity class. Therefore, by the SOP auction, w_i must be the first object $j \in \{j_{l+1}, \ldots, j_{l'}\}$ that does not belong to the similarity class $S(j_1)$. This is equivalent to saying

$$w_i = \max_{j \in A(i) \setminus S(j_i)} \{c_{ij} - p_j\}, \tag{3.3.59}$$

the definition given in the SO auction. Thus, the value of w_i given by $S(i)$ in the SOP auction is identical to the value of w_i given by i in the SO auction.

Each $S(i) \in \tilde{I}$ generates a bid in the SOP auction that is identical to the bid generated by $i \in \tilde{I}'$ in the SO auction. Therefore, the bidding phases are identical. Since the claims phases of the two auctions are exactly the same, the SO auction is a special case of the SOP auction, and the two are equivalent when $S(i) = \{i\}$ for all persons i. □

3.3.6.3 Proof for SOP auction is sufficient

Finally, I use my results about the relationships between the three algorithms to show that the proofs given by Bertsekas and Castañón for the SOP auction are sufficient to justify all three algorithms.

Corollary 3.3.9. *Any result which holds for the SOP auction algorithm holds for all three algorithms described as part of the extended auction method.*

Proof. Suppose some theorem holds for the SOP auction. The theorem must also hold for special cases of the SOP auction, such as the SO auction. Because the theorem holds for the SO auction, it also holds for special cases of the SO auction, such as the assignment auction algorithm. □

3.3.7 Considerations for the extended auction method

3.3.7.1 Explicit vs. implicit transformation

Even with integer data, a small transport problem may decompose into an assignment problem so large as to make the extended auction method untenable. To illustrate this, consider the following (admittedly degenerate) example:

Let $\{a,\, a+2\}$ and $\{b,\, b+2\}$ be two sets of twin primes. Choose $M = N = 2$, and

$$d_1 = a, \qquad d_2 = b + 2, \qquad s_1 = b, \qquad s_2 = a + 2.$$

Choose any negative-valued cost coefficients c_{ij} for $i = 1,\, 2$ and $j = 1,\, 2$. The transport problem is feasible, and its underlying transport graph has four vertices and four arcs. Now convert the transport problem to an assignment problem. Because the demand and supply quantities are relatively prime, it is impossible to "downsize" the resulting problem. This means the underlying assignment graph has $2(a + b + 2)$ vertices and $(a + b + 2)^2$ arcs.

Even if the twin prime conjecture turns out to be false, the largest known twin primes have more than $200\,000$ digits (in base-10 notation). Using these twin primes, a transport problem simple enough to solve by hand becomes an assignment problem requiring more vertices than the number of electrons in the universe. By comparison, an auction algorithm that did not require this transformation would require far less storage space. For example, the general auction, described in Chapter 4, would require no more than fourteen weight

variables[1]. Given an effective method for storing large integers, they would take up just over one megabyte of computer memory.

The example above motivates a consideration never directly addressed in [16]: the possibility of *implicit transformation*.

Definition 3.3.10. We say a sink i in $\mathcal{T}$ is *implicitly transformed* by an auction algorithm if the algorithm can be applied without creating and tracking the individual persons in $S(i)$. We can refer to this as an *implicit transformation.*

If an algorithm requires the creation of the individual persons in $S(i)$, one says i is *explicitly transformed* by that algorithm, or that the algorithm uses an *explicit transformation*.

Similar definitions apply for each source j in $\mathcal{T}$.

Note that it is possible to implicitly transform sources, sinks, or both. Of the three algorithms described for the extended auction method, one implicitly transforms only sinks and the other two rely on explicit transformation. It is worth describing these features of each algorithm in detail.

- The assignment auction explicitly transforms the sinks and sources of the transport problem.
- The SO auction explicitly transforms the sinks and sources of the transport problem.
- The SOP auction explicitly transforms the sources from the transport problem. Sinks are transformed implicitly, but the number of individual bids computed equals the number of persons that would have been generated by explicit transformation.

3.3.7.2 Rational values in the extended auction method

Optimal transport problems with rational costs and weights can be solved using the extended auction method by employing a scaling process.

[1]Two demand values d_i, two unsatisfied demand values D_i, two supply values s_j, two available supply values S_j, and up to six quantities q_{ij} (two for bidding and up to four for claim lists).

The cost-multiplying technique, described in Section 3.2.6.2, is not suggested in [16], but I suspect it would not have been surprising to the authors: the ε-scaling implementation they use multiplies all costs by constant multiples of $\min\{M, N\}$ so that $\varepsilon < 1$ always guarantees optimality [16, p. 91].

While working with rational-valued costs is relatively straightforward, handling the rational-valued weights in the extended auction algorithm becomes a bit trickier. Multiplying such weights by a common denominator in order to get integer values preserves the inherent weight relationships. Suppose all weights are multiplied by k, and that T^k is an optimal transport plan for the weight-multiplied problem. Then the transport plan

$$T = \left\{ \left(i,\, j,\, \frac{f_{ij}}{k} \right) \;\middle|\; (i,\, j,\, f_{ij}) \in T^k \right\} \tag{3.3.60}$$

is an optimal transport plan for the original problem, and the optimal primal cost for the original problem can be found by dividing the weight-multiplied solution by k.

This technique is not mentioned in [16], possibly for good reason.[2] When every weight is multiplied by common denominator k, the total weight of the problem is also multiplied by k, becoming kL. If the weight-multiplied problem is explicitly transformed into an assignment problem, it will have kL persons and kL objects. Depending on the sparsity of the underlying graph, it may have as many as $(kL)^2$ arcs. Thus, a rational-weighted transport problem with a common denominator that is even moderately large can generate an immense assignment problem. Theorem 3.2.8 implies that such problems could have horrendous worst-case complexity bounds (but see Section 3.3.5 for another possibility).

3.3.7.3 Irrational values in the extended auction method

In [16], Bertsekas and Castañón do not mention the possibility of using the extended auction algorithm on problems with irrational cost values. See Section 3.2.6.3 for some possible considerations implicit in the complexity argument given in [15].

[2] Weight-multiplying appears in [15, p. 259] during a discussion of the relaxation method.

By their nature, extended auction methods that explicitly transform sinks or sources cannot be applied where irrational weights are present. At best, those weights could be approximated by successively more precise rational values. However, the complexity issues stemming from the increasingly large common denominators would likely make such an approach prohibitive.

CHAPTER 4
THE GENERAL AUCTION

Here, I propose a more general auction method, one developed specifically for the transport problem (rather than the special-case assignment problem) and capable of handling real-valued costs and weights. To support this extension, I also consider how to determine error bounds and convergence rates with real-valued data. Finally, I compare auction methods using a series of standard problems. The key elements of this chapter have been submitted for publication; see [112].

4.1 General auction for the transport problem

The inspiration behind the general auction method is the same as that used for other auction methods: the real-world auction known as the open ascending price auction. However, the general auction method uses the transport problem as its basis, not the assignment problem. (Thus, the method's name: it is designed around a more general problem than other auction methods.) To avoid redundancy, I will define and explain only those terms and ideas which differ from the assignment auction method.

4.1.1 Description and terminology

Assume without loss of generality that $M \geq N$; that is, there are at least as many sinks as sources. Each sink i is a *bidder* in the auction, seeking to satisfy its demand d_i, and each source is a *lot* containing the supply s_j.

The general auction uses a variant form of lot bidding, similar to "times the money" bidding. Bidder i has unsatisfied demand D_i, so bidder i makes a bid of *bid amount* b_{ij} on lot j. The quantity desired from lot j is set to $q_{ij} = \min\{D_i,\ s_j\}$. One can write the bid as

the pair $(b_{ij},\, q_{ij})$. The actual bid is understood to be price b_{ij} per q_{ij} items, for a total bid value of $b_{ij}q_{ij}$.

Suppose that lot j has available supply S_j. If $q_{ij} \leq S_j$, the desired quantity is immediately available, so bidder i is awarded a *claim* on lot j of quantity q_{ij} at bid price b_{ij}. This claim, represented by the triple $(i,\, b_{ij},\, q_{ij})$, is added to the *claim list* for lot j, denoted by C_j.

Such claims can be "outbid"; that is, superseded by bids offering higher prices. If $q_{ij} > S_j$, compare bidder i's offer to those already on the claim list. The difference between q_{ij} and S_j is made up by taking the required amount from the lowest priced claim(s) with bid price less than b_{ij}. Only if insufficient low-priced claims exist will bidder i claim less than q_{ij}.

Even so, as long as b_{ij} is greater than the lowest bid price on lot j's current claim list, bidder i will be able to claim some quantity in lot j. To guarantee the occurrence of such a claim, for each lot j, determine a *lot price* p_j, defined as

$$p_j := \min_{(i,\, b_{ij},\, q_{ij}) \in C_j} \{b_{ij}\}. \tag{4.1.1}$$

(If C_j is empty, let p_j be equal to some initial price p_j^0.) When bidding, one requires that bid prices satisfy $b_{ij} > p_j$.

The lot price vector $\mathbf{p} = \{p_j\}_{j=1}^{N}$ corresponds to the price vector used in the dual profit equation. Thus, like the assignment auction, the general auction attempts to solve the dual problem. The general auction also uses ε-complementary slackness, defined as given in Section 3.2.3 for the assignment problem.

4.1.1.1 Hungry cannibals

The Hungry Cannibal rule, described below, is a modification to the process of handling claim lists. While technically optional, incorporating it adds relatively little overhead, and

when the range of weights is large it can greatly reduce the number of iterations required for convergence.

The Hungry Cannibal rule is motivated by a potential slow-down during the iterative process. In the bidding phase, each sink attempts to maximize the quantity it acquires. However, it is possible for a sink i to bid on a lot j where it already has a claim, and for the new claim to supersede some or all of the old one. When this happens, the old claim is "cannibalized" by the new one, and the quantity acquired by i is no longer maximal. (It may even be zero.) Call any claim that outbids an earlier claim by the same bidder a *cannibal* claim.

It is possible to speed up the bidding process, guaranteeing the maximum possible quantity is acquired by each bid, even when confronted by cannibal claims. One can deal with cannibals by implementing the *Hungry Cannibal rule*, or *HC rule*: during the claims phase, if a new claim by i cannibalizes a quantity q, that quantity is added to the quantity desired by the new claim. Thus, i can cannibalize itself during the claims phase, but the "hunger" of i remains maximal. As a side effect, the HC rule can unify adjacent bids by the same bidder in each claim list, helping to keep the size of the claim lists manageable.

The HC rule makes little difference in the auction results. Suppose the rule is not in place: assuming consistent sorting, i bids for lot j again on the following bidding phase, repeatedly if necessary, until i makes a claim on j that is not a cannibal. Thus, the main difference when applying the HC rule is a reduction in the number of iterations required. Since the general auction does not require the HC rule to function, the rule is technically optional. However, it requires relatively little expense to implement, considering the potential savings. The HC rule is clearly labeled as such in Step (1)-(b) of the claims phase for the general auction.

The proofs given below assume the HC rule is in place, ensuring that the quantity acquired by every new claim is maximal. This assumption, while not strictly necessary, helps simplify the arguments.

4.1.2 Iteration

To initialize the general auction method, one must have a bidding step size ε and initial lot price vector. Once initialized, the auction is performed in iterations. Each iteration consists of two phases: a bidding phase and a claims phase.

Bidding phase of the general auction: Let $\tilde{I}$ be a nonempty subset of sinks i that are unsatisfied under the current transport plan T. For each person $i \in \tilde{I}$:

(1) Find:

(a) The lot j_i offering best expense, given by

$$j_i = \arg\max_{j \in A(i)} \{c_{ij} - p_j\} \tag{4.1.2}$$

(b) The second-best expense, chosen by considering lots other than j_i,

$$w_i = \max_{j \in A(i),\, j \neq j_i} \{c_{ij} - p_j\} \tag{4.1.3}$$

If j_i is the only source in $A(i)$, define w_i to be $-\infty$. (For computational purposes, this can be any value satisfying $w_i \ll c_{ij_i} - p_{j_i}$.)

(2) Compute the bid $(b_{ij_i},\, q_{ij_i})$, where the bid price b_{ij_i} is given by

$$b_{ij_i} = c_{ij_i} - w_i + \varepsilon \tag{4.1.4}$$

and the quantity claimed q_{ij_i} is equal to

$$q_{ij_i} = \min\{D_i,\, s_{j_i}\}. \tag{4.1.5}$$

Claims phase of the general auction: For each source j, let I_j be the set of sinks from which j received a bid in the bidding phase of the iteration. If I_j is nonempty, for each $i_j \in I_j$ with bid $(b_{i_j j},\ q_{i_j j})$:

(1) While $S_j < q_{i_j j}$ and $p_j \leq b_{i_j j}$:

(a) Find the lowest-priced claim $c = (k,\ b_{kj},\ q_{kj})$ given by

$$c = \underset{(i,\, b_{ij},\, q_{ij}) \in C_j}{\arg\min} \{p_{ij}\}, \tag{4.1.6}$$

(b) If $k = i_j$, add q_{kj} to $q_{i_j j}$. [*HC rule*]

(c) Find the quantity in c to be claimed by bidder i_j,

$$q_{i_j c} = \min\{q_{i_j j},\ q_{kj}\}. \tag{4.1.7}$$

(d) Make the quantity $q_{i_j c}$ available:

(i) Add $q_{i_j c}$ to S_j.

(ii) Subtract $q_{i_j c}$ from q_{kj}.

(iii) If $q_{kj} = 0$, remove c from C_j and update p_j.

(iv) Add $q_{i_j c}$ to D_k.

(2) Let $q_{i_j j} = \min\{q_{i_j j},\ S_j\}$, and if $q_{i_j j} > 0$:

(a) Insert $(i_j,\ b_{i_j j},\ q_{i_j j})$ into C_j.

(b) Subtract $q_{i_j j}$ from D_{i_j}.

(c) Update p_j.

4.1.3 Solution

When all sinks have been satisfied, the general auction terminates. The resulting complete transport plan is equal to

$$T = \bigcup_{j=1}^{n} C_j \tag{4.1.8}$$

and the primal cost of T is equal to

$$\sum_{j=1}^{n} \sum_{(i,\, b_{ij},\, q_{ij}) \in C_j} c_{ij} q_{ij}. \tag{4.1.9}$$

If one wishes to represent T as a simplified transport plan, there is still one more step to perform. Using the claim lists, determine the simplified flow for each $(i,\, j)$ as

$$f_{ij} := \sum_{(i,\, b_{ij},\, q_{ij}) \in C_j} q_{ij}, \tag{4.1.10}$$

(If C_j does not contain a claim by bidder i, assume $f_{ij} = 0$.) Using these flow values, the simplified complete transport plan $\widetilde{T}$ equals

$$\widetilde{T} := \{\, (i,\, j,\, f_{ij}) \mid (i,\, j) \in \mathcal{A},\, f_{ij} > 0 \,\}. \tag{4.1.11}$$

The primal cost of the simplified transport plan equals

$$\sum_{(i,\, j) \in \mathcal{A}} c_{ij} f_{ij}, \tag{4.1.12}$$

exactly the form used in the transport problem.

4.2 Mathematical Results

The original proofs for the assignment auction and its extensions rely heavily on the integral nature of their transport problem data. In considering the mathematics underlying the general auction algorithm, assume transport problems with real-valued data.

Assuming real-valued data invalidates many of the standard assumptions for auction algorithms. Unlike Bertsekas and Castañón, one cannot assume that any quantities claimed are bounded away from zero. Thus, one must consider the possibility that claimed quantities tend to zero as the number of bids goes to infinity. One also cannot assume that lot

price increases are bounded away from zero, so one must consider the possibility that price increases tend to zero as the number of bids goes to infinity. Finally, given the definition of lot prices as a minimum, one cannot assume that lot prices change at all. One must consider the possibility that lot prices remain fixed over infinitely many bids.

Because of these considerations, I establish the properties of the general auction algorithm using a much different approach than that taken by Bertsekas and Castañón. The key result, Theorem 4.2.3, will effectively show that none of the above possibilities can occur; the general auction behaves well when applied to real-valued data, and terminates after a finite number of iterations. I then consider how the general auction is related to the other auction methods.

4.2.1 Termination and optimality of the general auction

4.2.1.1 General auction prices are nondecreasing

Theorem 4.2.1. *When applying the general auction method, lot prices are nondecreasing.*

Proof. Assume $\varepsilon > 0$ is given. Fix the lot j^* and let its lot price before and after iteration be given by p_{j^*} and p'_{j^*}, respectively. If no sink bids on j^*, then the lot price of j^* is unchanged, so $p'_{j^*} = p_{j^*}$. Suppose instead that some sink i bids on j^*, with bid price and claim $(b_{ij^*},\, q_{ij^*})$. From Equation (4.1.2), the expense of sink i's bid is given by

$$x_i = \max_{j \in A(i)} \{c_{ij} - p_j\} = c_{ij^*} - p_{j^*} \tag{4.2.1}$$

Thus, the bid price b_{ij^*} is given by

$$b_{ij^*} = c_{ij^*} - w_i + \varepsilon = p_{j^*} + x_i - w_i + \varepsilon. \tag{4.2.2}$$

By Equation (4.1.3), $x_i - w_i \geq 0$, so

$$b_{ij^*} = p_{j^*} + x_i - w_i + \varepsilon \geq p_{j^*} + \varepsilon > p_{j^*} \tag{4.2.3}$$

Since the new bid b_{ij^*} is at least as high as the current lot price p_{j^*}

$$p'_{j^*} \geq \min\{p_{j^*},\, b_{ij^*}\} \geq p_{j^*}. \tag{4.2.4}$$

Since this is true for all i that bid on j^*, it must be that $p'_{j^*} \geq p_{j^*}$. Therefore, lot prices are nondecreasing. □

4.2.1.2 Steady price implies satisfaction

Theorem 4.2.2. *If a bid by sink i on lot j does not increase the lot price p_j, then i becomes satisfied, and all other bidders that were satisfied without increasing the price p_j remain satisfied. If i should become unsatisfied, then i will bid on a lot offering the same expense that j did when it satisfied i.*

Proof. Fix the lot j^*. Let p_{j^*} be the price of lot j^* prior to the bid by i, and assume the second-highest bid price on the claim list C_{j^*} is

$$\hat{p}_{j^*} > p_{j^*}. \tag{4.2.5}$$

If $|C_j| < 2$, it is sufficient to assume $\hat{p}_{j^*} = +\infty$.

Suppose p'_{j^*} is the lot price of j^* at some later time, and that $p'_{j^*} = p_{j^*}$.

Assume first that there are no satisfied sinks on the claim list of j^*, and let i_1 be the sink currently bidding $(b_{i_1 j^*},\, q_{i_1 j^*})$ on j^*. Let q_1 be equal to

$$q_1 = \sum_{\{\,(k,\, b_{kj^*},\, q_{kj^*})\;\mid\; b_{kj^*} = p_{j^*}\,\}} q_{kj^*}. \tag{4.2.6}$$

Because the HC rule is in place, one can assume without loss of generality that $k \neq i_1$ for all claims of price p_{j^*}.

As shown in Equation (4.2.3), $b_{i_1 j^*} \geq p_{j^*} + \varepsilon$. Thus, if $q_1 \leq q_{i_1 j^*}$, all claims of price p_{j^*} have been overbid and the price after i_1's claim satisfies

$$p'_{j^*} \geq \min\{b_{i_1 j^*},\, \hat{p}_{j^*}\} > p_{j^*}. \tag{4.2.7}$$

This contradicts the supposition that $p'_{j^*} = p_{j^*}$, and so it must be that $q_1 > q_{i_1 j^*}$. By definition, $q_{i_1 j^*} = \min\{D_{i_1},\, s_{j^*}\}$. If $q_{i_1 j^*} = s_{j^*}$, then $q_1 > s_{j^*}$, which contradicts the definition of q_1 as a quantity claimed on C_{j^*}. Hence, $q_{i_1 j^*} = D_{i_1}$. Because i_1 claimed the quantity D_{i_1} and the HC rule is in place, it must be the case that sink i_1 is satisfied and

$$\hat{p}_{j^*} = \min\{b_{i_1 j^*},\, \hat{p}_{j^*}\} > p_{j^*}. \tag{4.2.8}$$

is preserved.

Now assume j^* has one other sink that was satisfied without increasing the price p_{j^*}. Without loss of generality, assume the sink is i_1. Suppose some sink i_2 bids $(b_{i_2 j^*},\, q_{i_2 j^*})$ on lot j^*. Because i_1 is satisfied, $i_2 \neq i_1$. The remaining quantity available at price p_{j^*} equals

$$q_2 = \sum_{\{\,(k,\, b_{kj^*},\, q_{kj^*})\;|\;b_{kj^*} = p_{j^*}\,\}} q_{kj^*}. \tag{4.2.9}$$

Because of the HC rule, one can assume $k \neq i_2$ for all claims of price p_{j^*}. (As the steps above show, it is also true that $k \neq i_1$ for all such claims.)

Once again, $b_{i_2 j^*} \geq p_{j^*} + \varepsilon$. Note that the bid price associated with the quantity claimed by i_1 exceeds p_{j^*}, so i_2 cannot claim any quantity from i_1 unless $q_2 \leq q_{i_2 j^*}$. However, in that case all claims of price p_{j^*} have been overbid and the price after i_2's claim satisfies

$$p'_{j^*} \geq \min\{b_{i_2 j^*},\, \hat{p}_{j^*}\} > p_{j^*}. \tag{4.2.10}$$

This contradicts the supposition that $p'_{j^*} = p_{j^*}$, and so one must have $q_2 > q_{i_2 j^*}$, and i_1 remains satisfied.

By definition, $q_{i_2 j^*} = \min\{D_{i_2},\, s_{j^*}\}$. If $q_{i_2 j^*} = s_{j^*}$, once again there is a contradiction because the total quantity in C_{j^*} exceeds s_{j^*}. Hence, $q_{i_2 j^*} = D_{i_2}$. Because i_2 claimed the quantity D_{i_2} and the HC rule is in place, it must be the case that sink i_2 is also satisfied and

$$\hat{p}_{j^*} > p_{j^*} \tag{4.2.11}$$

is preserved.

Continuing inductively, one finds that up to $M-1$ distinct bidders can be satisfied while maintaining $p'_{j^*} = p_{j^*}$, and that

$$\hat{p}_{j^*} > p_{j^*} \tag{4.2.12}$$

is preserved by each of them. (Because the quantity with price p_{j^*} must be owned by at least one bidder, it is not possible for more than $M-1$ distinct bidders to satisfy the initial assumption that $p'_{j^*} = p_{j^*}$.)

Suppose now that i_k becomes unsatisfied for some $k = 1, \ldots, M-1$. Because expenses are nonincreasing and $p'_{j^*} = p_{j^*}$, $x_{i_k j^*}$ must still offer the best expense for bidder i_k. Therefore, i_k bids on some lot with expense equal to $x_{i_k j^*}$. □

4.2.1.3 General auction terminates

Theorem 4.2.3. *Given $\varepsilon > 0$, if at least one feasible transport plan exists, then the general auction method terminates.*

Proof. Let I be the set of bidders (sinks) and J be the set of lots (sources) for a feasible transport problem $\mathcal{T}$. As a consequence of Theorem 4.2.1, one can partition J into four subsets:

- J^F, the set of lots which receive finitely many bids.

- J^M, the set of lots whose prices achieve maximum values, while receiving infinitely many bids.
- J^A, the set of lots whose prices asymptotically approach finite limits, without ever achieving those limits.
- J^∞, the set of lots whose prices increase without bound.

Now consider the sets

$$I^F = \{\, i \in I \mid i \text{ bids finitely many times } \} \tag{4.2.13}$$

$$I^M = \{\, i \in I \mid i \text{ bids on some } j \in J^M \text{ infinitely many times } \} \tag{4.2.14}$$

$$I^A = \{\, i \in I \mid i \text{ bids on some } j \in J^A \text{ infinitely many times } \} \tag{4.2.15}$$

$$I^\infty = \{\, i \in I \mid i \text{ bids on some } j \in J^\infty \text{ infinitely many times } \}. \tag{4.2.16}$$

I will show that the four sets partition I.

Suppose $i \in I$ such that $A(i) \setminus J^\infty$ is nonempty. Then there exists $j^* \in A(i) \setminus J^\infty$ such that p_{j^*} is bounded above. Thus, after a finite number of iterations

$$x_i = \max_{j \in A(i)} \{c_{ij} - p_j\} \geq c_{ij^*} - p_{j^*} > c_{ij^\infty} - p_{j^\infty} \qquad \forall\, j^\infty \in J^\infty. \tag{4.2.17}$$

Therefore, if $J^\infty \cap A(i)$ is nonempty, it must be that $i \notin I^\infty$. By the contrapositive,

$$A(i) \subseteq J^\infty \qquad \forall\, i \in I^\infty. \tag{4.2.18}$$

This implies that I^∞ is pairwise disjoint with I^F, I^A, and I^M.

Suppose there exists $i \in I \setminus I^F$ such that $J^A \cap A(i) \neq \varnothing$ and $J^M \cap A(i) \neq \varnothing$. By definition, i bids infinitely many times on lots from J^A and/or J^M. After a finite number of iterations k, all lots in J^F no longer receive bids and all lots in J^M have reached their fixed prices. Define

$$l_j := \lim_{k \to \infty} p_j^k \qquad \forall\, j \in J^A, \tag{4.2.19}$$

where p_j^k is the lot price of j at the end of the k-th iteration. Let

$$j^M = \underset{j \in A(i) \cap J^M}{\arg\max} \{c_{ij} - p_j\} \tag{4.2.20}$$

and

$$j^A = \underset{j \in A(i) \cap J^A}{\arg\max} \{c_{ij} - l_j\}. \tag{4.2.21}$$

One of two possibilities must exist:

1. If $c_{ij^M} - p_{j^M} \geq c_{ij^A} - l_{j^A}$, then for any iteration $\hat{k} \geq k$ one has $c_{ij^M} - p_{j^M}^{\hat{k}} > c_{ij^A} - p_{j^A}^{\hat{k}}$. Thus, for any $\hat{k}$-th iteration such that $\hat{k} \geq k$, i will not bid on lots in J^A, and so $i \notin I^A$.
2. If $c_{ij^M} - p_{j^M} < c_{ij^A} - l_{j^A}$, then there exists iteration $\hat{k} \geq k$ such that $c_{ij^M} - p_{j^M} < c_{ij^A} - p_{j^A}$. Because prices are nondecreasing, after the $\hat{k}$-th iteration, i will not bid on lots in J^M, and so $i \notin I^M$.

Therefore, I^M and I^A are pairwise disjoint. By definition, I^F must be pairwise disjoint with both I^M and I^A, because a sink cannot bid both finitely many and infinitely many times. Thus, all four sets are pairwise disjoint, and since their union is I they constitute a partition.

I will now consider possible elements in the three sets I^M, I^A, and I^∞, to show that these sets are in fact empty.

1. Suppose I^M is nonempty. After a finite number of iterations all prices in J^M are fixed and all bidders in I^M no longer bid on lots outside J^M. From that iteration on, by Theorem 4.2.2, each bidder $i \in I^M$ must be satisfied after its bid. Thus, some other bidder(s) must cause each $i \in I^M$ to become unsatisfied infinitely many times. Because the prices in J^M are fixed, each time i becomes unsatisfied it will rebid on a lot in J^M with the exact same price as the one it chose on its previous bid. Therefore, after a finite number of additional iterations, all $i \in I^M$ will only have claims in J^F and J^M, and when they become unsatisfied it is by bidding against each other for lots in J^M.

Assume this has all occurred by the $(k-1)$-th iteration. Let D^k be the total unsatisfied demand at the start of the k-th iteration,

$$D^k = \sum_{i \in I^M} D_i, \tag{4.2.22}$$

and let Q^k be the total supply available at the fixed prices:

$$Q^k = \sum_{j \in J^M} \sum_{\substack{(i, b_{ij}, q_{ij}) \in C_j \\ b_{ij} = p_j}} q_{ij}. \tag{4.2.23}$$

Assume without loss of generality that $Q^k = u^k D^k$. Because the prices for lots in J^M remain unchanged, $u^k > 1$. During the round, each unsatisfied bidder in I^M must bid on some lot in J^M, and because the prices of those lots do not decrease, all those bidders must be satisfied. Thus, the total quantity acquired by bidders in I^M during the k-th round must equal D^k. Since the prices of the lots do not increase, the claimed quantities must have been taken from Q^k, and so Q^k must have been reduced by D^k. Because the new amounts claimed could come only from previous claims by bidders in I^M, at the end of the round the unsatisfied demand, D^{k+1}, must equal D^k, and the ratio of the available quantity, given by u^{k+1}, must satisfy $u^{k+1} \leq u^k - 1$. Therefore, after $\lceil u^k \rceil$ rounds, the available quantity at the current prices must have been exhausted, and the price of some lot in J^M must have increased. This contradicts the definition of J^M, and therefore I^M must be empty.

2. Suppose I^A is nonempty. By the definition of J^A, for each $j_r \in J^A$, there exists an associated iteration k_r such that the price at the start of that iteration, $p_{j_r}^{k_r}$, satisfies

$$l_r - p_{j_r}^{k_r} < \varepsilon. \tag{4.2.24}$$

Let $k = \max_r k_r$.

Assume without loss of generality that by the start of iteration k all lots in J^F have also received their final bids. This implies that all bidders in I^A are now bidding exclusively on lots in J^A. Recall from Equation (4.2.3) that each new bid exceeds the current lot price by at least ε. Thus, after the k-th iteration, for all $i \in I^A$ and all $j \in J^A$, the new bid price b_{ij} must exceed l_j.

Let D^k be the total unsatisfied demand at the start of the k-th iteration,

$$D^k = \sum_{i \in I^A} D_i, \tag{4.2.25}$$

and let Q^k be the total supply available at prices not exceeding the asymptotic limits:

$$Q^k = \sum_{j \in J^A} \sum_{\substack{(i, b_{ij}, q_{ij}) \in C_j \\ b_{ij} \leq l_j}} q_{ij}. \tag{4.2.26}$$

Assume without loss of generality that $Q^k = u^k D^k$.

Let i be some unsatisfied bidder in I^A, bidding on $j \in J^A$. The bid price offered by i, by exceeding l_j, exceeds the price of all quantities claimed in Q^k. By the definition of J^A, the price of lot j does not exceed l_j. Thus, all bidders in I^A must be satisfied at the end of their bids. This implies that the claimed quantities must have been taken from Q^k, and that

$$Q^{k+1} = (u^k - 1)D^k. \tag{4.2.27}$$

Because no bidders outside I^A will bid again on lots in J^A, the bidders made unsatisfied by the new claims on Q^k must be members of I^A. Thus, at the end of the round the unsatisfied demand, D^{k+1}, must equal D^k and the ratio of the available quantity u^{k+1}, must satisfy $u^{k+1} \leq u^k - 1$. Therefore, after $\lceil u^k \rceil$ rounds, the quantity claimed at prices below asymptotic bounds must have been exhausted, and the price of some lot $j \in J^A$ exceeds l_j. This contradicts the definition of J^A, and therefore I^A must be empty.

3. Suppose I^∞ is nonempty. From Items 1 and 2, $I = I^F \cup I^\infty$. Let $i \in I^\infty$. Equation (4.2.18) implies

$$\min_{j \in A(i)} p_j \geq \min_{j \in J^\infty} p_j, \tag{4.2.28}$$

and since $p_j \to +\infty$ for all $j \in J^\infty$, it must be the case that

$$\min_{j \in A(i)} p_j \to +\infty. \tag{4.2.29}$$

Therefore,

$$x_i = \max_{j \in A(i)} \{c_{ij} - p_j\} \to -\infty. \tag{4.2.30}$$

This implies that after a finite number of iterations, each lot in J^∞ will be satisfied exclusively by the bidders in I^∞. Otherwise, some bidder in I^F would become unsatisfied infinitely many times, causing them to bid infinitely often. This would contradict the definition of I^F.

Furthermore, because the algorithm does not terminate, there must be at least one bidder in I^∞ that is not satisfied, while all bidders in I^F have been satisfied by the lots in J^F. It follows that the total demand of the bidders in I^∞ must be strictly larger than the total supply of the lots in J^∞. However, by Equation (4.2.18), the demand in I^∞ can only be satisfied by supply in J^∞. This contradicts the assumption that a feasible transport plan exists. Therefore, I^∞ must be empty.

Because I^M, I^A, and I^∞ are all empty, $I = I^F$. Therefore, the general auction algorithm must terminate after finitely many bids. □

4.2.1.4 Minimum price increase for general auction

Corollary 4.2.4. *Given a feasible transport problem $\mathcal{T}$, there exists a finite $k \in \mathbb{N}$ and $\delta > 0$ such that every k bids the price of a lot is guaranteed to change, and when a lot price changes it increases by at least δ.*

Proof. This follows from Theorem 4.2.3, specifically the conclusion that $J^M \cup J^A = \varnothing$. □

4.2.1.5 General auction preserves ε-CS

Theorem 4.2.5. *If a transport plan and lot price vector satisfy ε-CS for the general auction method at the start of an iteration, the same is true of the transport plan and lot price vector obtained at the end of that iteration.*

Proof. Suppose ε-CS holds at the start of the iteration. Fix the lot j^* and let its price before and after iteration be given by p_{j^*} and p'_{j^*}, respectively.

Suppose that sink i bids on lot j^* during the iteration, and the lot price of j^* changes as a result. By Equations (4.1.2) and (4.1.4)

$$b_{ij^*} = c_{ij^*} - w_i + \varepsilon, \tag{4.2.31}$$

which implies, by applying Equation (4.1.3),

$$c_{ij^*} - b_{ij^*} = w_i - \varepsilon = \max_{j \in A(i),\, j \neq j^*} \{c_{ij} - p_j\} - \varepsilon. \tag{4.2.32}$$

Equation (4.1.1) guarantees that $p'_{j^*} \leq b_{ij^*}$, so

$$c_{ij^*} - p'_{j^*} \geq c_{ij^*} - b_{ij^*} = \max_{j \in A(i),\, j \neq j^*} \{c_{ij} - p_j\} - \varepsilon. \tag{4.2.33}$$

By Theorem 4.2.1, $p'_j \geq p_j$ for all j. Thus,

$$c_{ij^*} - p'_{j^*} \geq \max_{j \in A(i), j \neq j^*} \{c_{ij} - p'_j\} - \varepsilon. \tag{4.2.34}$$

Because $c_{ij^*} - p'_{j^*} \geq c_{ij^*} - p'_{j^*} - \varepsilon$,

$$c_{ij^*} - p'_{j^*} \geq \max_{j \in A(i)} \{c_{ij} - p'_j\} - \varepsilon, \tag{4.2.35}$$

and ε-CS is satisfied.

Suppose that i has a claim on j^*, but the lot price p_{j^*} has not changed during the iteration. Because ε-CS held prior to the iteration and $p_j \leq p'_j$ for all j,

$$c_{ij^*} - p'_{j^*} = c_{ij^*} - p_{j^*} \geq \max_{j \in A(i)} \{c_{ij} - p_j\} - \varepsilon \geq \max_{j \in A(i)} \{c_{ij} - p'_j\} - \varepsilon. \tag{4.2.36}$$

Thus, ε-CS holds for all claims in every claim list.

Therefore, ε-CS holds for all $(i,\ j,\ q_{ij})$ in the transport plan T. □

4.2.1.6 General auction error bound

Theorem 4.2.6. *Let $\varepsilon > 0$ be the minimum price increase of the general auction algorithm. If at least one feasible transport plan exists, then when the assignment auction method terminates, the resulting feasible transport plan is within $L\varepsilon$ of optimal, where*

$$L = \sum_{i=1}^{M} d_i = \sum_{j=1}^{N} s_j. \tag{4.2.37}$$

Proof. Assume the transport problem is feasible. Let P^* be the optimal primal solution to the transport problem,

$$P^* = \max_{(i,j) \in \mathcal{A}} c_{ij} f_{ij}, \tag{4.2.38}$$

and D^* be the optimal dual solution

$$D^* = \min_{p = \{p_j\}_{j=1}^{N}} \left\{ \sum_{i=1}^{M} d_i \max_{j \in A(i)} \{c_{ij} - p_j\} + \sum_{j=1}^{N} s_j p_j \right\}. \tag{4.2.39}$$

Let T^* be the simplified transport plan obtained when the auction terminates, and let $\mathbf{p}^* = (p_1^*, \ldots, p_N^*)$ be the resulting price vector. Let i be any sink and suppose $(i, j_r, f_{ij_r}) \in T^*$ for $j_1, j_2, \ldots, j_t$. Because $(T^*, \mathbf{p}^*)$ satisfies ε-CS,

$$\max_{j \in A(i)} \{c_{ij} - p_j^*\} - \varepsilon \leq c_{ij_r} - p_{j_r}^*. \tag{4.2.40}$$

Therefore, by rearranging and summing terms for all j_r,

$$\max_{j \in A(i)} \{c_{ij} - p_j^*\} + p_{j_r}^* \leq \varepsilon + c_{ij_r} \tag{4.2.41}$$

$$f_{ij_r}(\max_{j \in A(i)} \{c_{ij} - p_j^*\} + p_{j_r}^*) \leq f_{ij_r}(\varepsilon + c_{ij_r}) \tag{4.2.42}$$

$$\sum_{\substack{(i, j_r, f_{ij_r}) \in T^* \\ r=1, \ldots, t}} f_{ij_r}(\max_{j \in A(i)} \{c_{ij} - p_j^*\} + p_{j_r}^*) \leq \sum_{\substack{(i, j_r, f_{ij_r}) \in T^* \\ r=1, \ldots, t}} (f_{ij_r}\varepsilon + f_{ij_r}c_{ij_r}) \tag{4.2.43}$$

$$d_i \max_{j \in A(i)} \{c_{ij} - p_j^*\} + \sum_{\substack{(i, j_r, f_{ij_r}) \in T^* \\ r=1, \ldots, t}} f_{ij_r}p_{j_r}^* \leq d_i\varepsilon + \sum_{\substack{(i, j_r, f_{ij_r}) \in T^* \\ r=1, \ldots, t}} f_{ij_r}c_{ij_r}. \tag{4.2.44}$$

Summing over all sinks i, this gives

$$\sum_{i=1}^{M} d_i \max_{j \in A(i)} \{c_{ij} - p_j^*\} + \sum_{(i, j, f_{ij}) \in T^*} f_{ij}p_j^* \leq \sum_{i=1}^{M} d_i\varepsilon + \sum_{(i, j, f_{ij}) \in T^*} f_{ij}c_{ij}. \tag{4.2.45}$$

Given any sink j, $(i_u, j, f_{i_uj}) \in T^*$ for $u = 1, 2, \ldots, v$. Thus, summing first over the i_u for j, and then over all j,

$$\sum_{\substack{(i_u, j, f_{i_uj}) \in T^* \\ u=1, \ldots, v}} f_{i_uj}p_j^* = s_jp_j^* \tag{4.2.46}$$

$$\sum_{(i, j, f_{ij}) \in T^*} f_{ij}p_j^* = \sum_{j=1}^{N} s_jp_j^*. \tag{4.2.47}$$

Substituting this into Equation (4.2.45),

$$\sum_{i=1}^{M} d_i \max_{j \in A(i)} \{c_{ij} - p_j^*\} + \sum_{j=1}^{N} s_j p_j^* \leq \sum_{i=1}^{M} d_i \varepsilon + \sum_{(i,j,f_{ij}) \in T^*} f_{ij} c_{ij} \tag{4.2.48}$$

Therefore,

$$P^* = D^* \tag{4.2.49}$$

$$P^* \leq \sum_{i=1}^{M} d_i \max_{j \in A(i)} \{c_{ij} - p_j^*\} + \sum_{j=1}^{N} s_j p_j^* \tag{4.2.50}$$

$$P^* \leq \sum_{i=1}^{M} d_i \varepsilon + \sum_{(i,j,f_{ij}) \in T^*} f_{ij} c_{ij} \tag{4.2.51}$$

$$P^* \leq L\varepsilon + P^* \tag{4.2.52}$$

$$D^* \leq L\varepsilon + D^*. \qquad \square$$

4.2.2 Essential characteristics of the general auction

In *Network Optimization: Continuous and Discrete Models*, Dimitri Bertsekas describes what he considers the "important ingredients" of auction methods [15]. Here, in the same form used by Bertsekas, are what I consider the essential elements of the general auction method:

(a) ε-CS is maintained.

(b)(1) During each iteration, at least one bidder with unsatisfied demand claims supply in one lot.

(b)(2) The bid price of this claimed supply is increased by at least $\beta\varepsilon$, where $\varepsilon > 0$ and β is some fixed positive constant.

(b)(3) Any previously-claimed supply that is needed to satisfy a higher priced claim (if any) becomes unclaimed.

(c)(1) No bid price is decreased.

(c)(2) Any supply that was claimed at the start of an iteration remains claimed at the end of that iteration (although the bidder claiming it may change).

With the exception of (a), all of these characteristics are essential to the argument that the general auction algorithm terminates after a finite number of iterations. Characteristic (a) relates the the resulting price vector to the optimal price vector and establishes a worst-case bound on the distance from optimality.

4.2.3 SO auction is a special case of the general auction

Given an integer-valued transport problem, one can relate the general auction method to the extended auction method. This relationship bypasses the SOP auction algorithm, and establishes the SO auction as a special case of the general auction.

Theorem 4.2.7. *If $\mathcal{T}$ is a feasible integer-valued transport problem and $d_i = 1$ for all sinks i in $\mathcal{T}$, then the general auction algorithm is equivalent to the auction algorithm for similar objects.*

Proof. Let $\mathcal{T}$ be any feasible integer-valued transport problem such that for all sinks i, $d_i = 1$. Thus, for purposes of the SO auction one has $S(i) = \{i\}$ for all sinks i. The lot represented by $S(i)$ must be unsatisfied if and only if there exists exactly one person for the SO auction that is unsatisfied. Consider the bidding phase for such a person i.

If the object j_i offers best expense for the SO auction, then the lot represented by $S(j_i)$ also offers best expense, so the object j_i chosen by the SO auction corresponds to the choice of lots in the general auction. Let j' be the object such that for the SO auction

$$w_i = \max_{j \in A(i) \setminus S(j_i)} \{c_{ij} - p_j\} = c_{ij'} - p_{j'}. \tag{4.2.53}$$

As a consequence, it must be that

$$p_{j'} = \min_{j \in S(j')} p_j, \tag{4.2.54}$$

and so the second-best expense computed by the general auction method must equal that computed by the SO auction. Therefore, the bid prices for the SO and general auction must be equal. Because $\mathcal{T}$ is an integer-valued transport problem, $S(i)$ is unsatisfied, and $d_i = 1$, the quantity desired by the general auction must be

$$q_{ij_i} = \min\{D_i,\, s_{j_i}\} = 1. \tag{4.2.55}$$

Therefore, the two bidding phases are equivalent.

Let j be some object that receives one or more bids during the claims phase, and let i_j be the person that made the highest bid on j.

Suppose the object j has already been claimed by some person k. Because j is the lowest-priced object in $S(j)$, and each bid increases the price of an object by at least ε, this implies that the lot $S(j)$ is satisfied. The SO auction claim on object j corresponds to making a claim on the lot $S(j)$, and

$$b_{i_j S(j)} = c_{i_j S(j)} - w_{i_j} + \varepsilon \geq p_j + x_{i_j} - w_{i_j} + \varepsilon \geq p_j + \varepsilon. \tag{4.2.56}$$

Thus, in the general auction the lowest priced claim on the current claim list $C_{S(j)}$ will be removed.

Assume without loss of generality that the order of the claims in the claim list of the general auction matches the sorted order of the expenses determined in the SO auction algorithm. Then the lowest-priced claim corresponds to the object claimed by person k. As shown in the bidding phase, the quantity claimed by i is 1, so both methods add 1 to D_k. Removing the claim $(k,\, b_{kS(j)},\, 1)$ from the claim list $C_{S(j)}$ is equivalent to removing $(k,\, j,\, 1)$ from T.

Appending $(i_j,\, j,\, 1)$ to T is equivalent to inserting $(i_j,\, b_{i_j S(j)},\, 1)$ into $C_{S(j)}$, and both methods subtract 1 from D_{i_j}. The price update for the general auction is the same as

determining the lowest-priced object in $S(j)$ after the price increase on object j. Thus, the two lot phases are equivalent, and so the two auction algorithms are equivalent. □

4.2.4 Ramifications of general auction equivalence

With the equivalence shown in Theorem 4.2.7, the assignment auction is a special case of the general auction. Furthermore, many of the theorems given for the assignment auction become consequences of those proved for the general auction. For the assignment problem, one can show that Corollary 4.2.4 holds for $k = 1$ and $\delta = \varepsilon$. That result, combined with Theorem 4.2.1, is sufficient to establish Theorem 3.2.1, and Theorem 3.2.2 follows from Theorem 3.2.1. The termination argument given in Theorem 3.2.3 is a special-case of that given in Theorem 4.2.3, because integer data makes the emptiness of the sets J^A and J^M self-evident there. The ε-CS preservation result in Theorem 3.2.4 is a consequence of Theorem 4.2.5, and Theorem 3.2.5 is simply Theorem 4.2.6 with

$$L = \sum_{j=1}^{N} s_j = \sum_{j=1}^{N} 1 = N. \tag{4.2.57}$$

4.3 Implementation

4.3.1 Implementation strategies

Because the auction may require a large number of iterations, efficient implementation is vital to optimal performance. Here I share my insights for effective computation.

4.3.1.1 Bidders and bidding

Note that the definition of sources and sinks is arbitrary with respect to the transport solution. Because each bidder makes only one bid per iteration, the method runs most effectively when one chooses the larger set for the sinks, thus maximizing the number of

bidders. It is convenient to store d_i and D_i for each bidder i. It may also be helpful for i to have a copy of the subset of the cost vector given by $\{ c_{ij} \mid j \in A(i) \}$.

Each bidder needs access to a subset of the cost coefficients and the lot price vector, but cost coefficients are constant and lot prices are not altered during the bidding phase. Thus, bidding can be performed in parallel, using a separate process for each bidder.

For each bidder and each bidding phase, determining x_i and w_i requires a fresh sort of

$$\{c_{ij} - p_j, \forall j \in A(i)\}. \tag{4.3.1}$$

Because only the two largest members are needed, the ideal implementation determines only those, rather than sorting the entire list. One effective way to do this by applying the `quickselect` algorithm to find the second-largest expense, w_i. As a result, the largest expense x_i is also identified. While the `quickselect` algorithm has worst-case quadratic complexity, the worst case is highly unlikely, and `quickselect` has average-case linear complexity. Hence, this partial sorting technique will usually find x_i and w_i in linear time. Since one needs both x_i and its associated lot number, j_i, it is best to perform the partial sort on paired data structures of the form $(j,\ c_{ij} - p_j)$.

4.3.1.2 Lots and claiming

For fast and easy access, it is convenient to store s_j and S_j. It is also convenient to store individual claims as triple data structures of the form $(i,\ p_{ij},\ q_{ij})$, exactly as shown in the description.

Each lot needs access only to its individual claim list, and the bids offered to it. Since bids themselves need not be changed during the claims phase, claims can be performed in parallel, using a separate process for each lot. (In order to obtain objective results, parallel computation was not used in my numerical tests.)

One of the advantages of claim lists is that they can scale in response to the relative complexity of the transport problem. Take advantage of this property by constructing claim lists as dynamically sized data structures and growing them as needed. This implementation can save time as well as memory, because it reduces the size of the list that each lot must update during its claims phase.

At any given time, only the lowest-priced claim on the claim list needs to be considered. Thus, it is most efficient to store each claim list as a binary min heap, sorted by price. Worst-case complexity for such a heap is logarithmic for insertion and deletion, and constant when retrieving the lowest-priced claim.

4.4 Numerical results

My implementation of the general auction method was written in C++, and relies heavily on the C++11 Standard Library. I implemented four additional methods for testing: the assignment auction method, the extended auction SO and SOP algorithms, and the network simplex method. The auction implementations used in these tests are based on software I later released to the public: the open-source `AUCTION ALGORITHMS IN C++` project [111]. The assignment and extended auction methods were used for benchmarking and comparison, while the network simplex method was used to test solutions for optimality.

I wrote, compiled, and tested all the programs myself, in order to minimize possible confounds in my results. Because the extended auction method involves three distinct algorithms, I implemented and tested all three. As shown below, my implementation of the SOP auction algorithm generated the most favorable data on non-assignment problems, so I focused on comparisons involving that algorithm.

All of the methods were implemented to solve the general transport problem as described in Section 2.2.1, applying only those data restrictions necessary to satisfy the minimal requirements of the algorithm. One could improve on my results for any particular method, simply by customizing the code to handle specific purposes or environments.

However, my goal was to evaluate the average-case effectiveness of the underlying methods themselves, aside from any potential time savings due to specialized design.

The transport problems I used for testing were initialized by creating assignment problems with NETGEN [93]. The only modifications to the NETGEN code were increased array sizes (to generate the large problems desired) and alterations to the I/O (input/output) routines. The network generation code was not altered. When different weights or costs were desired, the existing nodes and arcs were modified using the random uniform distribution functions from the C++11 Standard Library.

4.4.1 Comparison of auction methods for assignment

When Bertsekas and Castañón compared their implementation of the extended auction to the performance of the assignment auction on standard assignment problems [16, p. 92], they found that the additional overhead required by the extended auction measurably slowed computation time. Thus, I wished to see how the performance of the general auction compared to that of the assignment and extended auction methods when applied to assignment problems of increasing size. The comparison done by Bertsekas and Castañón in 1989 used problems with 150 to 500 pairs of sources and sinks. Given the improvements in computation power since that time, my comparison starts at 3000 sinks and ends at 10000 (with an equal number of sources at each size). Like Bertsekas and Castañón, the number of arcs in each problem is 12.5% of the maximum possible. The cost range is also relevant, as it influences complexity scaling [15, p. 34]. For the cost range, I used a fixed value $C = 100$. The resulting times are given in Table 4.1.

Over all tested problem sizes, the computed times for the assignment and general auction methods are nearly identical, differing by less than 2%. The largest difference in computed time was less than two-tenths of one second. This suggests that the overhead required for the implementation of claim lists scales appropriately with relatively simple problems such as assignment (see Theorem 3.3.8 and Section 4.2.4).

Table 4.1: Time in seconds for assignment scaling
N sinks and N sources, $N^2/8$ arcs

N	Assignment auction	General auction	Extended auction	Extended (unoptimized)
3000	1.23	1.21	1.71	9.06
4000	2.65	2.68	3.52	18.63
5000	4.73	4.82	6.23	35.33
6000	6.96	7.06	8.82	60.55
7000	8.90	8.97	10.85	90.82
8000	12.60	12.64	15.98	132.93
9000	17.00	16.83	20.73	193.30
10000	18.88	19.05	23.04	249.03

The graph in Figure 4.1 displays the times visually. Only three lines are clearly visible, because at this resolution the assignment and general auction lines overlap considerably. Nonetheless, the graph offers a useful visual comparison of the scaling properties of all methods when applied to the assignment problem.

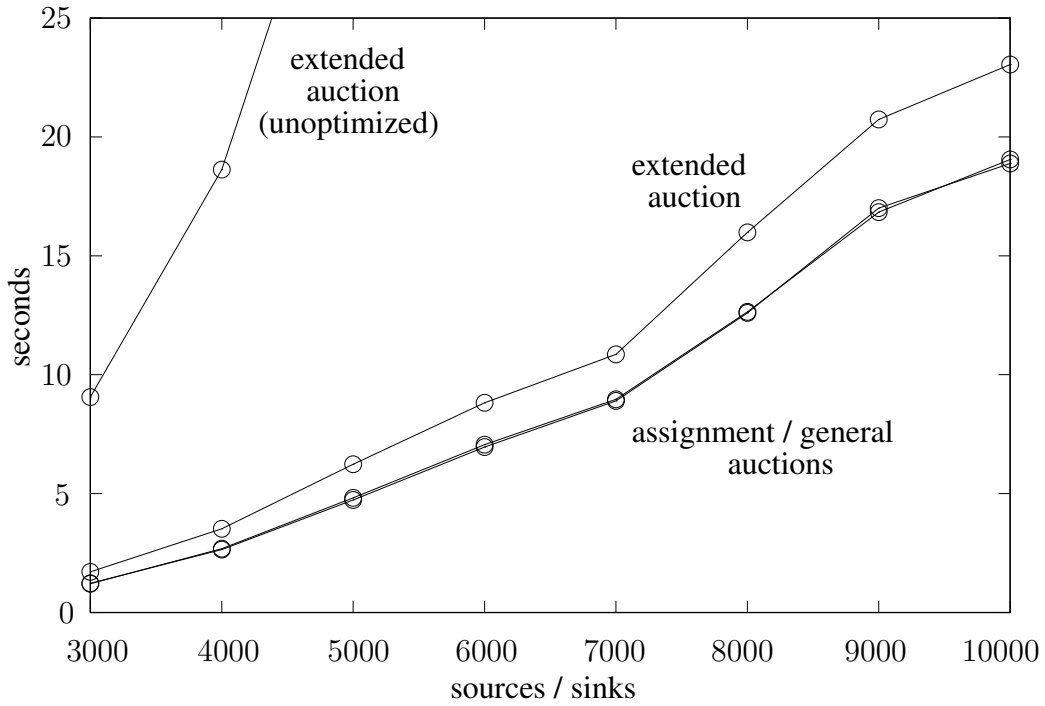


Figure 4.1: Solution time for assignment scaling
N sinks and N sources, $N^2/8$ arcs

Storage requirements were dominated by the need to store an explicit cost value for each arc, and so were nearly identical for all three algorithms. Storage scaled quadratically with respect to the number of sinks. Each assignment problem required approximately $7.1 \times 10^{-6}\, N^2$ megabytes of memory.

4.4.2 Comparison of extended and general auction

As described in Section 3.3.7.1, the explicit transformation of data used by most of the algorithms of the extended auction method makes them vulnerable to transport problems with arbitrarily large total weights. The SOP auction implementation I created uses implicit transformation for sinks, but explicit transformation for sources, making it an ideal vehicle for examining this vulnerability. I generated a small transport problem and scaled the total weight, comparing the time required by the extended and general auction methods.

In every case, the structure of the transport problem was identical. It had 500 sources, 500 sinks, and the same 225000 arcs (90% of maximum). The cost range was $C = 100$. Using a uniform integer distribution, the weight was spread among the sources and sinks. Each was given a minimum weight of one, but no maximum was assumed. The total weights and computation times for the two algorithms are given in Table 4.2.

The data collected suggests that, as weight increases, time for the SOP auction algorithm scales on the order of $\mathcal{O}(L^3)$, where L is the total weight of the transport problem. Time increases for the general auction algorithm as well, but it appears to be scaling sublinearly. The storage requirements for the SOP auction appear to scale linearly. This is not surprising, given that the SOP auction algorithm explicitly transforms the set of sinks into L persons. Storage requirements for the general auction are approximately constant.

The time increases given by both algorithms suggest that the weight range, or number of possible weight values, may be relevant to complexity calculations. The impact of increasing weight was not considered in [16], and of course is irrelevant to the assignment problem, but it seems natural to consider it. The larger the weight range, the more likely

Table 4.2: Results for weight-only scaling
500 sinks and 500 sources, 225000 arcs

(a) Time in seconds

Weight L	General auction	Extended auction
500	0.073	0.089
600	0.094	3.580
700	0.095	13.803
800	0.099	7.233
900	0.126	8.674
1000	0.132	8.316
2000	0.234	20.319
3000	0.348	106.755
4000	0.307	105.078
5000	0.441	178.798

(b) Storage in megabytes

Weight L	General auction	Extended auction
500	10.78	10.72
600	10.79	11.78
700	10.80	12.81
800	10.81	13.84
900	10.81	14.88
1000	10.82	15.91
2000	10.88	26.28
3000	10.97	36.70
4000	11.00	47.02
5000	11.08	57.54

it becomes that each sink will have positive flows over multiple arcs, increasing the complexity of the optimal transport plan, and thus the time required to calculate it.

Given the degree to which weight increases adversely affect the performance of the extended auction method, I also compared the scaling behaviors of the two methods in problems where the relative total weight was fixed. I randomly generated transport problems with N sources and N sinks, where the total weight of each problem was $L = 2N$. The number of arcs for each problem was fixed at 90% of maximum, and the cost range was $C = 100$. The results are shown in Table 4.3.

Table 4.3: Results for fixed weight ratio scaling
N sinks and N sources, $0.9N^2$ arcs, total weight $2N$

(a) Time in seconds

N	General auction	Extended auction
500	0.13	8.32
600	0.19	37.28
700	0.31	63.27
800	0.42	177.86
900	0.61	233.96
1000	0.81	291.29
1100	1.08	920.57

(b) Storage in megabytes

N	General auction	Extended auction
500	10.82	15.91
600	20.15	27.53
700	25.23	35.26
800	30.77	43.86
900	36.48	53.05
1000	42.89	63.36
1100	70.52	95.38

For these problems, the general auction method appears to scale with respect to time at $\mathcal{O}(N^3)$, whereas the time scaling for the extended auction seems to be $\mathcal{O}(N^6)$. Because of the need to store an explicit cost value for each arc, both methods scale quadratically with respect to storage. The extended auction consistently requires approximately 150% of the amount of storage needed for the general auction method. The additional storage is used to explicitly transform supply nodes into objects.

4.4.3 General auction performance on real-valued transport

To test the performance of the general auction algorithm on real-valued data, I generated transport problems using the following method:

- Sources and sinks are points in the plane, restricted to the unit square $[0,\ 1] \times [0,\ 1]$. Points are placed randomly using the uniform distribution, with a guarantee of some minimal spacing between points in the same set, to ensure that cost differences do not become too small for machine precision. I generate N sinks and an equal number of sources.
- The graph underlying the transport problem is the complete bipartite graph $K_{N,\,N}$, so the problem is maximally dense with N^2 arcs. The cost function is the Euclidean distance, so given sink $i = (x_i,\ y_i)$ and sink $j = (x_j,\ y_j)$, the cost of the arc connecting them is
$$c_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{4.4.1}$$
- The weights of the N sinks are initially given the N values $\sqrt{1},\ \sqrt{2},\ \ldots,\ \sqrt{N}$, randomly distributed among the points. The weights of the sources are generated using the same method. Because the total weights of the sinks equals the total weight of the sources, and the transport graph is complete, the problem is guaranteed to be feasible.

I chose this method because it populates the weights and costs with irrational values, and there is minimal repetition within the sets. The resulting problems are not pathological, but they certainly are not simple.

More importantly, the large number of unique irrational values generated by this method addresses the heart of the concern with real-valued transport problems: whether irrational-valued data impacts the average-case complexity of the general auction method.

Given the spatial restriction of the unit square, the absolute maximum cost for each of these problems is bounded above by $\sqrt{2}$. However, because the data is not limited to the integers, the range of cost and weight values is significantly larger than the maximum absolute cost would indicate. In fact, the range of cost and weight values significantly exceeded N for every problem I considered.

I found that initializing ε with twice the absolute maximum cost generated good results. Using that initial ε and a zero-vector of initial prices, I solved each problem by iterating through successive ε-scaling phases until the optimal solution was achieved.

As described above, it is difficult to determine a priori minimum ε values that guarantee optimal solutions. To account for this, I confirmed the optimality of my solutions a posteriori, by using the resulting transport network as an initial network for the network simplex method. The amount of computation done by the network simplex method, and the final cost it obtained, allowed me to determine how close the general auction solutions were to optimal. If the general auction generated a sub-optimal result for that value of ε, I discarded that result and recomputed the solution using a smaller ε value.

Because the total weights of the problems were normalized, Theorem 4.2.6 guarantees that the solutions are at most ε from optimal. Given the limits of machine precision, it is possible for the general auction to achieve the same optimal cost obtained by the network simplex, even if the transport network generated is in fact sub-optimal.

To consider the case where an optimal transport network was desired, after achieving the optimal cost I continued to increase the number of iterations until my network simplex

solver indicated that no pivot operations needed to be performed. Using this technique, I forced the general auction method to run until both its primal cost and its transport plan were as good as what could be achieved using the network simplex method. I judged these solutions to be as good as could be achieved, given the limits of machine precision. The results of these tests are given in Table 4.4.

Table 4.4: Real-valued general auction results
N sinks and N sources, N^2 arcs

N	Time (sec)	ε-scaling phases	Minimum ε value
500	1.06	9	1.05×10^{-5}
1000	6.42	9	1.05×10^{-5}
2000	41.90	10	2.63×10^{-6}
3000	104.54	11	6.62×10^{-7}
4000	214.99	11	6.67×10^{-7}

My code used long integers and double-precision floating point numbers. Because of the numerical methods used, I assumed that the calculations would be accurate up to the square root of machine precision. For my machine, this limit was equal to

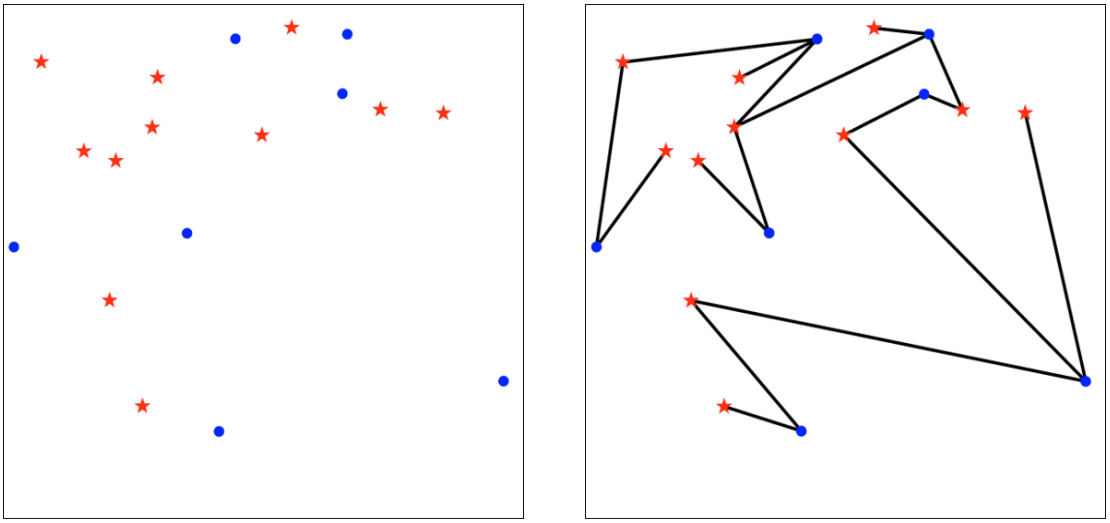
$$\sqrt{\texttt{eps}} := 2^{-26} \approx 1.490116 \times 10^{-8}. \tag{4.4.2}$$

For problem sizes above $N = 4000$, the ε value achieved this level of precision without generating a completely optimal transport plan, as judged by the network simplex method.

Not surprisingly, given the dense and complicated nature of these sample transport problems, the time complexity resembles $\mathcal{O}(N^3)$. My implementation of the general auction algorithm generates the full array of arc costs, so storage complexity grows quadratically with respect to N. (Had I wished, I could have taken advantage of the structure of the problems to instantiate cost values as needed.) Because my previous examples use floating point storage, the results presented in Table 4.3 give a representative sample of the storage size progression.

4.4.4 A specific example

Consider the specific example shown in Figure 4.2(a). The graph is embedded in $[0,\ 1]^2$, and the underlying network is assumed to be a bipartite complete graph. The points in X, $\{\mathbf{x}_i\}_{i=1}^{7}$, are located at the blue dots, and the points in Y, $\{\mathbf{y}_j\}_{j=1}^{11}$, are located at the red stars. μ and ν are uniform, so for each $i \in \mathbb{N}_7$, $\mu(\mathbf{x}_i) = 1/7$ and for each $j \in \mathbb{N}_{11}$, $\nu(\mathbf{y}_j) = 1/11$.



(a) Points embedded in $[0,\ 1]^2$ (b) Optimal solution

Figure 4.2: Discrete problem embedded in $[0,\ 1]^2$

The positions of the points were approximated using the (scaled) problem shown on the right side of Figure 2 in [5]. The exact coordinates I used are given in Table 4.5.

The problem can be solved directly using the general auction method. By multiplying the common denominator 77, the problem weights can be converted to integers. This allows a solution to be computed using the other three auction methods. For all computations, the step size $\varepsilon = 2 \times 10^{-8}$ was used. Since the total mass is $L = 1$, that means the Wasserstein distance approximation error is bounded above by $\varepsilon = 2 \times 10^{-8}$.

Table 4.5: Positions of discrete points in Figure 4.2

Points in X	Points in Y
$\mathbf{x}_1 = \left(\frac{50}{2327}, \frac{1203}{2327}\right)$	$\mathbf{y}_1 = \left(\frac{167}{2327}, \frac{2023}{2327}\right)$
$\mathbf{x}_2 = \left(\frac{822}{2327}, \frac{1265}{2327}\right)$	$\mathbf{y}_2 = \left(\frac{359}{2327}, \frac{1629}{2327}\right)$
$\mathbf{x}_3 = \left(\frac{966}{2327}, \frac{388}{2327}\right)$	$\mathbf{y}_3 = \left(\frac{471}{2327}, \frac{967}{2327}\right)$
$\mathbf{x}_4 = \left(\frac{1036}{2327}, \frac{2123}{2327}\right)$	$\mathbf{y}_4 = \left(\frac{503}{2327}, \frac{1586}{2327}\right)$
$\mathbf{x}_5 = \left(\frac{1514}{2327}, \frac{1879}{2327}\right)$	$\mathbf{y}_5 = \left(\frac{619}{2327}, \frac{500}{2327}\right)$
$\mathbf{x}_6 = \left(\frac{1536}{2327}, \frac{2145}{2327}\right)$	$\mathbf{y}_6 = \left(\frac{663}{2327}, \frac{1735}{2327}\right)$
$\mathbf{x}_7 = \left(\frac{2236}{2327}, \frac{609}{2327}\right)$	$\mathbf{y}_7 = \left(\frac{687}{2327}, \frac{1952}{2327}\right)$
	$\mathbf{y}_8 = \left(\frac{1153}{2327}, \frac{1701}{2327}\right)$
	$\mathbf{y}_9 = \left(\frac{1287}{2327}, \frac{2173}{2327}\right)$
	$\mathbf{y}_{10} = \left(\frac{1683}{2327}, \frac{1811}{2327}\right)$
	$\mathbf{y}_{11} = \left(\frac{1963}{2327}, \frac{1799}{2327}\right)$

In [5], the exact solution is given as 0.2615. Using the points in Table 4.5, the general auction gives an approximate Wasserstein distance of

$$W_1 \approx 0.2615134292833 \pm 7.802 \times 10^{-10}.$$

The approximation and error bound can be obtained from the primal cost and ε, respectively. However, in this case a better approximation and error bound was obtained by contrasting the primal and dual solutions,

$$W_1 \approx \frac{\widetilde{D}^* + \widetilde{P}^*}{2} \pm \frac{|\widetilde{D}^* - \widetilde{P}^*|}{2}, \tag{4.4.3}$$

which one can always use as an *a posteriori* worst-case estimate.

Comparing the individual solutions offered by the four auction methods is also instructive. The primal solution given by the general auction is

$$\widetilde{P}^*_{\mathrm{GA}} = 0.2615134300634707013.$$

As it turns out, the primal solutions of the four methods differ from each other by no more than 3.896×10^{-17}. However, there is a significant difference between the dual solutions of the four algorithms. Table 4.6 shows the primal-dual difference, $|\widetilde{D}^* - \widetilde{P}^*|$, along with the time and memory required by each of the four methods.

Table 4.6: Results of embedded problem comparison

Auction	$\lvert\widetilde{D}^* - \widetilde{P}^*\rvert$	time (sec)	storage (KB)
Assignment	7.258×10^{-10}	1.012×10^{-2}	158.4
SO	3.375×10^{-3}	$8.504 \times 10^{+1}$	158.4
SOP	1.289×10^{-1}	1.798×10^{-3}	27.40
General	1.560×10^{-9}	1.480×10^{-3}	14.50

Noteworthy features appear in all three columns of Table 4.6: the primal-dual difference, time, and storage.

First, consider the primal-dual difference. For some reason, the SO and SOP auctions give significantly worse dual results than the general and assignment auctions. The key feature of the SO and SOP auctions is that they alter the way that dual prices are computed, in order to more quickly approach the correct solution. Perhaps, in the process, the SO and SOP methods reduce the accuracy of those same dual prices.

Now look at the time values. The time requirements for the assignment, SOP, and general auction are unsurprising. The SOP auction, since it minimizes bidding wars, is notably faster than the assignment auction. The general auction, which avoids bidding wars entirely, is even faster. The SO auction, on the other hand, is thousands of times slower than any of the other methods. This is consistent with earlier numerical results, suggesting that the SO auction's method of minimizing bidding wars, when taken alone, may cause more delays than it prevents.

Finally, consider the values for storage. Even for such a small problem, the storage difference between explicit and implicit expansion is visible in the storage values. The general auction, which does not expand any part of the problem, requires only 14.50 KB of memory. The SOP auction, which partially expands the problem, requires almost twice as

much memory: 27.40 KB. The assignment and SO auctions, which both fully expand the problem, require nearly eleven times as much memory as the general auction: 158.4 KB.

CHAPTER 5
THE BOUNDARY METHOD

5.1 Introduction

In this chapter, I present a new solution method for optimal transport problems over product spaces that are semi-discrete: one of the measures is discrete and the other is continuous. While semi-discrete formulations can be used to approximate solutions to fully continuous problems, the semi-discrete optimal transport problem is of practical relevance itself. The fundamental elements of this chapter have been submitted for publication; see [44].

5.1.1 Semi-discrete problem

The semi-discrete optimal transport problem is the Monge-Kantorovich problem of Definition 2.1.1, with restrictions on μ and ν, and c:

1. Assume that μ satisfies the following:
   (a) μ is bounded.
   (b) μ is nonatomic.
   (c) μ is continuous except on a set of Lebesgue measure zero.
   (d) The support of μ is contained in the convex and compact region $A \subseteq X$.
2. Assume ν has exactly n non-zero values, located at $\{\mathbf{y}_i\}_{i=1}^n \subseteq Y$.
3. Assume c is an admissible ground cost function, as described in Definition 5.1.2, below.

The restrictions on μ and ν are inherent in the problem's definition, while the admissibility restrictions on c were chosen to facilitate the definition of a broad class of important semi-discrete transport problems.

Remark 5.1.1. Without loss of generality, henceforth assume $n \geq 2$. If $n = 1$, the optimal transport solution is trivial: $T(\mathbf{x}) \equiv \mathbf{y}_1$ for all $\mathbf{x} \in X$, and

$$P^* = \int_A c(\mathbf{x},\, \mathbf{y}_1)\, d\mu(\mathbf{x}). \tag{5.1.1}$$

Definition 5.1.2 (Admissible ground cost)**.** An *admissible ground cost* function is a measurable, continuous, function $c(\mathbf{x}_1,\, \mathbf{x}_2) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, satisfying the following properties:

1. $c(\mathbf{x},\, \mathbf{x}) = 0$ for all $\mathbf{x} \in \mathbb{R}^d$;
2. $c(\mathbf{x}_1,\, \mathbf{x}_2) = c(\mathbf{x}_2,\, \mathbf{x}_1) > 0$ for all $\mathbf{x}_1,\, \mathbf{x}_2 \in \mathbb{R}^d$ such that $\mathbf{x}_1 \neq \mathbf{x}_2$;
3. $\|\mathbf{x}_1 - \mathbf{x}_2\|_2 \leq \|\mathbf{x}_3 - \mathbf{x}_2\|_2$ implies $c(\mathbf{x}_2,\, \mathbf{x}_1) \leq c(\mathbf{x}_2,\, \mathbf{x}_3)$ for all collinear points $\mathbf{x}_1,\, \mathbf{x}_2,\, \mathbf{x}_3 \in \mathbb{R}^d$.

For all c and all $\mathbf{x} \in \mathbb{R}^d$, $S \subseteq \mathbb{R}^d$, define

$$c(\mathbf{x},\, S) := \inf_{\mathbf{s} \in S} c(\mathbf{x},\, \mathbf{s}). \tag{5.1.2}$$

Remark 5.1.3. The semi-discrete problem has interesting applications in its own right. Recent developments include work on the semi-discrete principal-agent problem and systems of unequal dimension; see [45, 36].

There are also obvious similarities between the semi-discrete problem and (generalized) Voronoi diagrams, similarities that are clearly recognized in [104]. In the simplest formulation of the Voronoi problem, one has n points $\mathbf{y}_i$'s in a planar region R, and seeks a partition of the given region into sub-regions R_i's so that each of them contains exactly one $\mathbf{y}_i$, and every point in R_i is closer to $\mathbf{y}_i$ than to any other point $\mathbf{y}_j$, $j \neq i$. The most commonly used distance is the Euclidean distance, in which case the regions are polygons.

Voronoi diagrams play an important role in many applications: epidemiology, medicine (bone, cell, and muscle structures), computing atomic charges in chemistry, development of autonomous navigation systems in robotics, computer graphics, geophysics, meteorol-

ogy, and computational methods, particularly mesh smoothing and computational geometry [94].

Computing Voronoi diagrams in the plane is a standard problem, with hundreds of algorithms for their construction. For the planar case, Fortune's sweepline algorithm is one highly efficient technique, scaling as $\mathcal{O}(N \log N)$, with respect to the number of non-zero values of ν. [56]. In higher dimensions, and for non-Euclidean distances, the problem can still be challenging.

These similarities notwithstanding, the semi-discrete optimal transportation problem is considerably more general, as it controls for both distance and region volume. In addition, my numerical method departs from any used to compute Voronoi cells.

5.1.1.1 Semi-discrete transport and the Monge problem

Because c is continuous and μ is nonatomic, there is at least one solution to the semi-discrete Monge-Kantorovich problem that also satisfies the Monge problem, as described in Definition 2.1.4 (see [98]). Thus, by applying Equation (2.1.11), one can assume without loss of generality that any transport plan π partitions A into n sets A_i, where A_i is the set of points in A that are transported by the map T to $\mathbf{y}_i$. Using this partitioning scheme in combination with Equation (2.1.12) one can rewrite the primal cost function for the semi-discrete problem as

$$P(\pi) := \sum_{i=1}^{n} \int_{A_i} c(\mathbf{x},\, \mathbf{y}_i)\, d\mu(\mathbf{x}), \tag{5.1.3}$$

and the dual cost function for the semi-discrete problem as

$$D(\varphi,\, \psi) := \int_A \varphi\, d\mu + \sum_{i=1}^{n} \psi(\mathbf{y}_i). \tag{5.1.4}$$

5.1.2 Shift characterization for semi-discrete optimal transport

This idea of sets A_i underlies the shift characterization of the semi-discrete optimal transport problem. The definition of the characterization, which follows, is based on one given by Rüschendorf and Uckelmann in [104, 102].

Definition 5.1.4 (Shift characterization)**.** Let $\{a_i\}_{i=1}^n$ be a set of n finite values, referred to as *shifts*. Define

$$F(\mathbf{x}) := \max_{1 \leq i \leq n} \{a_i - c(\mathbf{x}, \mathbf{y}_i)\}. \tag{5.1.5}$$

For $i \in \mathbb{N}_n$, where $\mathbb{N}_n = \{1, \ldots, n\}$, let

$$A_i := \{\mathbf{x} \in A \mid F(\mathbf{x}) = a_i - c(\mathbf{x}, \mathbf{y}_i)\}. \tag{5.1.6}$$

Note that $\cup_{i=1}^n A_i = A$.

As given in [101, 104], the sets $\{A_i\}_{i=1}^n$ define an optimal transport plan if and only if

$$\forall\, i \in \mathbb{N}_n, \qquad \mathbf{x} \in A_i \implies \pi^* : \mathbf{x} \to \mathbf{y}_i \tag{5.1.7}$$

on sets of positive μ-measure.

Note that Equation (5.1.7) does not describe the amount of mass transported from $\mathbf{x}$ to $\mathbf{y}_i$. If the shifts $\{a_i\}_{i=1}^n$ define F so that $\mu(A_i) = \nu(\mathbf{y}_i)$ for all $i \in \mathbb{N}_n$, then

$$\sum_{n=1}^{n} \mu(A_i) = \mu(A), \tag{5.1.8}$$

and the shift characterization determines a Monge solution on sets of positive μ-measure. In this case, an optimal transport plan π^* can be determined by simply identifying the appropriate shifts $\{a_i\}_{i=1}^n$. Because there is a transport map T associated with the Monge solution that satisfies

$$\mathbf{x} \in A_i \implies T(\mathbf{x}) = \mathbf{y}_i \qquad \forall\, i \in \mathbb{N}_n \tag{5.1.9}$$

except on a set of μ-measure zero, one says that the semi-discrete transport problem is Monge under the shift characterization. A formal description is given in Definition 5.3.8.

For some problems, Equation (5.1.8) is not satisfied by the appropriate set of shifts. In this case, $\mu(A_i) \geq \nu(\mathbf{y}_i)$ for all $i \in \mathbb{N}_n$, with strict inequality for at least one i. The solution determined is not Monge, and a linear program must be solved to determine the amount of mass that the optimal plan π^* transports from each $\mathbf{x}$ to the appropriate $\mathbf{y}_i$. The general structure of this linear program is described in Remark 5.3.12. If Equation (5.1.8) is not satisfied, one says that the semi-discrete transport problem is not Monge under the shift characterization. See Remark 5.3.11 for additional details.

Remark 5.1.5. The shift characterization has seen some application in the literature of numerical optimal transport. Of particular note is the work [104], where Rüschendorf and Uckelmann report on numerical experiments with ground costs given by the Euclidean distance taken to the powers 2, 3, 4, and 10. They assume that μ is the uniform distribution, and test various weights and placements for the set $\{\mathbf{y}_i\}_{i=1}^n$. When an exact solution cannot be directly determined, their approach consists in fully discretizing the problem and then using a LP solver (see Section 2.4.1). Rüschendorf and Uckelmann are well aware of the large computational expense of this approach, which may explain why they restrict approximation of solutions to one-dimensional distributions and problems in $\mathbb{R}^2$ with relatively few $\mathbf{y}_i$'s ($n \leq 15$). In [104], Rüschendorf and Uckelmann do not discuss approximating the Wasserstein distance, even for problems where an exact optimal transport map is known.

The shift characterization is also discussed in [5].[1] This paper of Barrett and Prigozhin proposes and focuses on a new numerical method for the optimal transport problem with ground cost given by the Euclidean distance. Starting with an alternative form of Equation (2.3.3), taken from [22], Barrett and Prigozhin develop a mixed formulation of the Monge-Kantorovich problem, which they solve using a finite element discretization. In the numerical examples, a single set of five $\mathbf{y}_i$'s is given, solved exactly with $a_i \equiv 0$ for all i,

[1] In the research of Barrett and Prigozhin, shift characterizations are referred to as "optimal couplings."

and then approximated with μ and ν both uniform distributions (continuous and discrete, respectively). Aside from the complexity of Barrett and Prigozhin's approach, including the difficulties inherent in the mesh selection process and the sensitive limiting process in the regularization parameter, they are not able to adequately resolve the region boundaries, which are at best blurred (see [5, Figure 3]). Like Rüschendorf and Uckelmann, Barrett and Prigozhin's paper does not concern itself with the Wasserstein distance.

5.2 Boundary Method

Here, I introduce my new method: the *boundary method*. At a high level, the idea of the method is simple: to track exclusively the boundaries between regions, without resolving the regions' interiors. In order to do this in practice, and to obtain an efficient technique, the interplay between discretization, a mechanism for discarding interior regions, and a fast solver, must all be accounted for.

5.2.1 Boundary identity and system of equations

For all $i,\ j \in \mathbb{N}_n$ such that $i \neq j$, let

$$A_{ij} := A_i \cap A_j. \tag{5.2.1}$$

The *boundary set* is defined as

$$B := \bigcup_{1 \leq i < n} \bigcup_{i < j \leq n} A_{ij}, \tag{5.2.2}$$

and for each $i \in \mathbb{N}_n$, let the *strict interior* of A_i is defined as

$$\mathring{A}_i := A_i \setminus B. \tag{5.2.3}$$

For all $i,\ j \in \mathbb{N}_n$ such that $i \neq j$, define $g_{ij} : X \to \mathbb{R}$ as

$$g_{ij}(\mathbf{x}) := c(\mathbf{x},\ \mathbf{y}_i) - c(\mathbf{x},\ \mathbf{y}_j). \tag{5.2.4}$$

By Corollary 5.3.16 below, B is nonempty, and for each $\mathbf{x} \in B$, there exist $i,\ j \in \mathbb{N}_n$, $i \neq j$, such that $\mathbf{x} \in A_{ij}$. Because $\mathbf{x} \in A_i$, $F(\mathbf{x}) = a_i - c(\mathbf{x},\ \mathbf{y}_i)$, and because $\mathbf{x} \in A_j$, $F(\mathbf{x}) = a_j - c(\mathbf{x},\ \mathbf{y}_j)$. Combining and rearranging these two equations, one finds that

$$g_{ij}(\mathbf{x}) = a_i - a_j\ , \qquad \forall \mathbf{x} \in A_{ij}. \tag{5.2.5}$$

Thus, Equation (5.2.5) implies that A_{ij} is a subset of a level set of g_{ij}; the value $a_i - a_j$ is constant, regardless of which $\mathbf{x} \in A_{ij}$ is chosen. Using this information, for each $i,\ j \in \mathbb{N}_n$, $i \neq j$, such that $A_{ij} \neq \varnothing$, define the constant *shift difference*

$$a_{ij} := g_{ij}(\mathbf{x}_{ij})\ , \qquad \text{for every } \mathbf{x}_{ij} \in A_{ij}. \tag{5.2.6}$$

Given a sufficiently large set of functionally independent equations of the form given in Equation (5.2.6), one could determine most or all of the shifts $\{a_i\}_{i=1}^n$. As Theorem 5.3.18 shows, it is possible to obtain exactly $(n-1)$ functionally independent equations of the desired form, but a set of n such independent equations does not exist.

The inherent lack of uniqueness of the shifts $\{a_i\}_{i=1}^n$ is intuitively apparent in the definitions of the sets A_i and the function F. While transforming the shifts from a_i to $(a_i + \sigma)$ alters the value of F (to $F + \sigma$), the change has no impact whatsoever on the composition of the sets A_i.

Since the set of shifts allows exactly one degree of freedom, the boundary method's approach is to obtain $(n-1)$ well-chosen a_{ij} values, fix one a_i, and use functionally independent equations of the form given in Equation (5.2.6) to solve for the remaining $(n-1)$

shifts. The crucial observation is that for the a_i's, there is no need to retain information about interior of the regions.

Remark 5.2.1. Also the Wasserstein distance can be computed without needing to save region interiors. If one has determined that $R \subset \mathring{A}_i$ for some region R, the (partial) Wasserstein distance corresponding to R is equal to

$$P_R := \int_R c(\mathbf{x},\, \mathbf{y}_i)\, d\mu(\mathbf{x}), \tag{5.2.7}$$

and the total Wasserstein distance P^* is equal to the sum of all such partial distances P_R, computed over all A_i.

Recognizing these facts, inherent in the shift characterization, inspired both the boundary method's name and its guiding principles, summarized below:

> Do *not* approximate the entire transport plan over A; rather, identify approximate region boundaries and determine $(n-1)$ shift differences a_{ij}.

5.2.2 The Boundary Method

As described below, generate a grid A^r approximating the unevaluated region of A, and use it to determine the subgrid B^r containing the boundary set B. This subgrid is determined by solving an approximated optimal transport problem from the grid A^r to the point set $\{\mathbf{y}_i\}_{i=1}^n$.

For convenience, this work restricts itself to $A = [0, l]^d$ (see Section 5.3.4.1), and applies a Cartesian grid over that region. At the r-th refinement level of the algorithm, the grid will thus consist of a collection of boxes with width w_r in each dimension of the discretization. By a slight abuse of notation, use $\mathbf{x}^r$ to refer to such a box, centered at the point $\mathbf{x}$. Thus, $\mu(\mathbf{x}^r)$ refers to the μ-measure of the box of width w_r centered at $\mathbf{x}$.

Neighboring boxes are those with center points that differ by no more than one unit in any discretization index. The set of *neighbors* of $\mathbf{x}$ is denoted $N(\mathbf{x})$ (defined fully in Equation (5.3.23), below). Because regions of μ-measure zero need not be transported to any particular $\mathbf{y}_i$, boxes of positive weight that are adjacent to such regions are always retained; that is Refer to such a box as an *edge box*. The set of edge boxes is defined as

$$\mathrm{edg}(A^r) := \{\mathbf{x} \in A^r \mid \mu(\mathbf{x}) > 0 \ \text{ and } \ \mu(\mathbf{x}_n) = 0 \ \text{ for some } \ \mathbf{x}_n \in N(\mathbf{x})\}. \tag{5.2.8}$$

Because A contains the support of μ, every box of positive mass that is adjacent to the boundary of A is an edge box.

A box whose neighbors and itself all have positive measure is referred to as an *internal box*. The set of internal boxes is

$$\mathrm{int}(A^r) := \{\mathbf{x} \in A^r \mid \mu(\mathbf{x}) > 0 \ \text{ and } \ \mu(\mathbf{x}_n) > 0 \ \text{ for all } \ \mathbf{x}_n \in N(\mathbf{x})\}. \tag{5.2.9}$$

Boxes of μ-measure zero are not part of $\mathrm{edg}(A^r)$ or $\mathrm{int}(A^r)$, and they are discarded when the optimal transport problem is solved. One need not be concerned about losing a region A_i due to this discard process, since this can only happen if $\mu(A_i) = 0$ (hence $\nu(\mathbf{y}_i) = 0$, which is not possible).

Region interiors are identified by comparing the destination of each $\mathbf{x} \in \mathrm{int}(A^r)$ to the destinations of its neighbors. Edge boxes are never considered part of the region interior, so they are passed directly to B^r.

In order to remove identified region interiors, one must also maintain a running total of the untransported mass, given by *partial measure* $\tilde{\nu}$. In order to maintain the balance of the transport problem, each time a region $\mathbf{x}^r$ is transported from A to $\mathbf{y}_i$, the remaining amount that can be transported to $\mathbf{y}_i$, $\tilde{\nu}(\mathbf{y}_i)$, must be reduced by $\mu(\mathbf{x}^r)$. The details are shown in Algorithm 5.1.

Algorithm 5.1: Boundary method

Boundary method algorithm

(0) Set $\widetilde{P} = 0$, $\tilde{\nu} = \nu$, and $r = 1$. Create $A^r = A^1$, the first discretization, from A.

(1) ***Quickly approximate*** the discretized transport solution.

(2) For each $\mathbf{x} \in \mathrm{int}(A^r)$:

Are the neighbors of $\mathbf{x}$ all transported to the same $\mathbf{y}_i$?

- If so, then $\mathbf{x}^r$ is in the interior of A_i:
  - [optional] Add $\int_{\mathbf{x}^r} c(\mathbf{z},\, \mathbf{y}_i)\, d\mu(\mathbf{z})$ to $\widetilde{P}$.
  - Reduce the value of $\tilde{\nu}(\mathbf{y}_i)$ by $\mu(\mathbf{x}^r)$.
  - Remove $\mathbf{x}$ from $\mathrm{int}(A^r)$.

The sets $\mathrm{edg}(A^r)$ and the reduced set $\mathrm{int}(A^r)$ form B^r, the boundary set for iteration r.

(3) Is the desired refinement reached?

- If not:
  - Refine B^r to create A^{r+1}, increment r, and go to Step (1).

Once the desired refinement level is reached:

(4) Use B^r to identify $(n-1)$ appropriate shift differences $\{a_{ij}\}$

and solve for the shifts $\{a_i\}_{i=1}^n$.

(5) [optional] Use $\widetilde{P}$ and B^r to approximate P^*.

Remark 5.2.2. Using the idea presented in Remark 5.2.1, one can approximate the Wasserstein distance P^* by generating a running total over region interiors: $\widetilde{P}$. As defined here, $\widetilde{P}$ is an increasing function of r, and for all r, $P^* \geq \widetilde{P}$. The Wasserstein distance over any remaining boundary region is evaluated at completion.

Remark 5.2.3. Obviously, some further approximations may be required to make the above a true general algorithm. Depending on μ, it may be necessary to approximate the mass of each box, $\mu(\mathbf{x}^r)$. Depending on μ and c, the Wasserstein distance over each box, given by

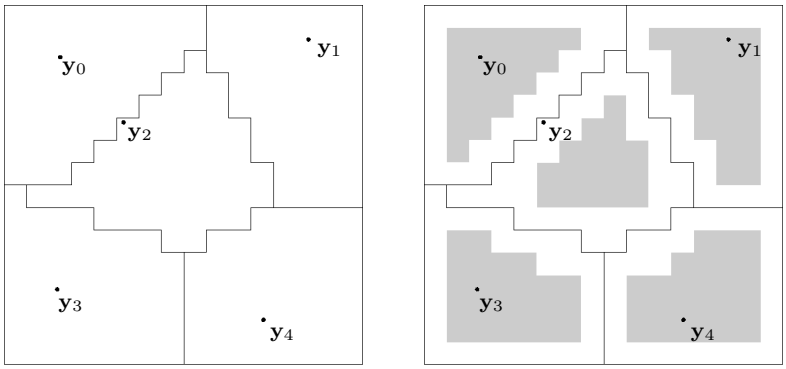
$\int_{\mathbf{x}^r} c(\mathbf{z},\, \mathbf{y}_i)\, d\mu(\mathbf{z})$, may also require approximation. Look to Section 5.4 for a discussion of such considerations.

To illustrate the iterative portion of the boundary method, Steps (1) and (2), consider the following example, based on the one Barrett and Prigozhin presented in [5].

Example 5.2.4. Let $X = Y = [0,\, 1]^2$. Assume μ is the uniform probability density, so for all measurable sets $S \subseteq A$, $\mu(S) = |S|$, and that ν has uniform discrete probability density, so $\nu(y_i) = 1/n$ for $1 \leq i \leq n$. Take $n = 5$, with the five points where ν has nonzero density distributed as shown in Figure 5.1.

Let c be the squared Euclidean norm, $\|\mathbf{y} - \mathbf{x}\|_2^2$. Suppose a discretization with width 2^{-5} is sufficient to provide the desired accuracy and that one applies the boundary method with initial width 2^{-4}.

Figure 5.1 shows the state of the boundary method during the first iteration. In Figure 5.1(a), Step (1) has just been completed : the approximate transport map has been computed, but the algorithm has not yet identified interior points or added anything to the partial transport cost $\widetilde{P}$. Figure 5.1(b) shows the state of the algorithm after Step (2): the interior regions have been identified (shown in gray), and the partial transport cost has been computed for those regions, giving $\widetilde{P} = 0.01387$.



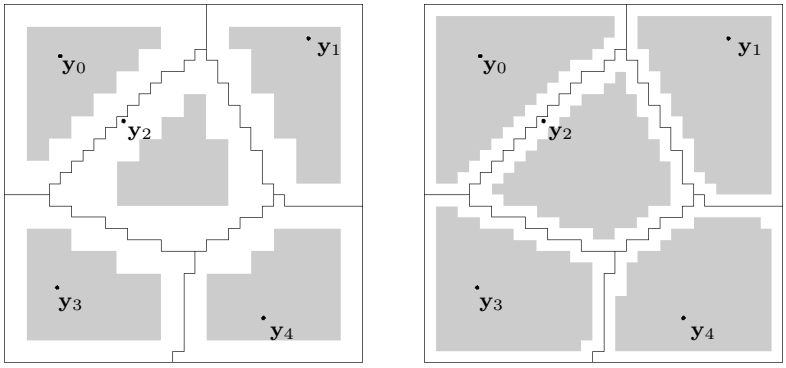
(a) A^1 after Step (1): $\widetilde{P} = 0.00000$

(b) A^1 after Step (2): $\widetilde{P} = 0.01387$

Figure 5.1: Iteration A^1 of Example 5.2.4 illustrated: $w_1 = 2^{-4}$

Figure 5.2 shows the state of the boundary method algorithm during the second iteration. In Figure 5.2(a), Step (1) has just been completed. As can be seen by comparing Figure 5.1(b) to Figure 5.2(a), the boundary and interior regions are the same ones that were present at the end of the first iteration, but refining the boundary set to width $w_2 = 2^{-5}$ allows one to compute a more refined transport map. Because Step (1) does not add to the identified interior regions, the partial Wasserstein distance $\widetilde{P}$ is also unchanged from Figure 5.1(b).

After Step (2) of the second iteration, shown in Figure 5.2(b), more of the interiors have been identified. The partial transport cost shows a corresponding increase: now $\widetilde{P} = 0.02898$. Because this is the desired refinement, a width of 2^{-5}, move on to Step (4), ending the iterative process.



(a) A^2 after Step (1): $\widetilde{P} = 0.01387$

(b) A^2 after Step (2): $\widetilde{P} = 0.02898$

Figure 5.2: Iteration A^2 of Example 5.2.4 illustrated: $w_2 = 2^{-5}$

Next, I discuss the key elements of the algorithm: Steps (1), (4) and (5).

5.2.2.1 Solving the discrete optimal transport solution: the general auction algorithm

Step (1) of the algorithm clearly emphasizes the characteristics required of an ideal transport solver for the boundary method: it needs only to approximate the correct solution, since misdirected points will be included in B^r, and their associated regions will be assigned after the next grid refinement. Moreover, computing the approximation must be *fast*,

and, for obvious reasons, it needs to possess reasonable scaling properties. To satisfy these requirements, and to bypass the shortcomings of standard LP solvers (see Section 2.4.1), I have turned my attention to the distributed relaxation methods known as auction algorithms; see [15] and [17].

For the boundary method, I apply my new auction algorithm, the *general auction*. As described in Chapter 4, the general auction offers significant performance advantages over other auction algorithms. Public domain C++ software implementing the general auction can be found on the Internet at [111].

5.2.2.2 Computing individual shift differences

Once one has reached the desired level of refinement for the boundary, one needs to use the set B^r to identify $(n-1)$ shift differences a_{ij}.

Suppose one has chosen to compute a specific shift difference a_{ij}. (The issue of choosing i and j is discussed in Section 5.2.2.3 below.) Step (1) determined a map T showing which $\mathbf{x} \in B^r$ were transported to $\mathbf{y}_i$. Step (2) used T to consider which of the neighbors of those points $\mathbf{x}$ were transported to $\mathbf{y}_j$. Thus, one can create the set of unordered pairs

$$A^r_{ij} := \{\, \{\mathbf{x}_i,\, \mathbf{x}_j\} \in B^r \mid T(\mathbf{x}_i) = \mathbf{y}_i,\ T(\mathbf{x}_j) = \mathbf{y}_j, \ \text{ and } \ \mathbf{x}_j \in N(\mathbf{x}_i)\,\}. \tag{5.2.10}$$

If i and j were well-chosen, such that $A_{ij} \neq \varnothing$, then by Theorem 5.3.34 (below), it must be that $A^r_{ij} \neq \varnothing$.

As shown in Equation (5.2.6), all $\mathbf{x}_{ij} \in A_{ij}$ give the identical exact value of a_{ij}. Hence, one can use any pair $\{\mathbf{x}_i,\, \mathbf{x}_j\} \in A^r_{ij}$ to compute an *approximate shift difference*

$$\tilde{a}_{ij} := g_{ij}\left(\frac{\mathbf{x}_i + \mathbf{x}_j}{2}\right) \tag{5.2.11}$$

Because there are almost certainly many points in A^r_{ij}, the approximation of a_{ij} is generally significantly overdetermined.

The goal is to use well-chosen points from a subset of A_{ij}^r, to bound a_{ij} without creating a situation where the approximation has no solution. Do this by focusing on approximate *intersection points*: points where the set A_{ij} intersects either the boundary of A or a third region A_k (where $k \neq i$ and $k \neq j$). These approximate intersection points are generated from the unordered pairs in

$$I_{ij}^r := \left\{ \{\mathbf{x}_i, \mathbf{x}_j\} \in A_{ij}^r \;\middle|\; \begin{array}{l} \mathbf{x}_i \text{ or } \mathbf{x}_j \in \mathrm{edg}(A) \;\text{ or} \\ \exists\, k \neq i,\, j,\; \mathbf{x}_k \in N(\mathbf{x}_i) \cup N(\mathbf{x}_j) \;\text{ s.t. }\; T(\mathbf{x}_k) = \mathbf{y}_k \end{array} \right\}. \tag{5.2.12}$$

In practice, I have always found $\left|I_{ij}^r\right| \geq 2$. However, even if $I_{ij}^r = \varnothing$ for some choice of μ, ν, and c, one can fall back on computing a_{ij} by setting $I_{ij}^r = A_{ij}^r$.

For each $\{\mathbf{x}_i, \mathbf{x}_j\} \in I_{ij}^r$, use Equation (5.2.11) to compute an approximate shift difference a_{ij}. One can also compute bounds, based on the range of possible values of g_{ij} on the set

$$\mathbf{x}_{ij}^r := \mathbf{x}_i^r \cup \mathbf{x}_j^r. \tag{5.2.13}$$

These bounds are

$$M\left(\{\mathbf{x}_i, \mathbf{x}_j\}\right) := \sup_{\mathbf{x} \in \mathbf{x}_{ij}^r} g_{ij}(\mathbf{x}) \qquad \text{and} \qquad m\left(\{\mathbf{x}_i, \mathbf{x}_j\}\right) := \inf_{\mathbf{x} \in \mathbf{x}_{ij}^r} g_{ij}(\mathbf{x}). \tag{5.2.14}$$

By choosing a single element at random from I_{ij}^r, one can determine $\tilde{a}_{ij}$ using Equation (5.2.11). The error bound on this estimate $\tilde{a}_{ij}$ is given by

$$\alpha_{ij} := \max\{M(\mathbf{x}_i, \mathbf{x}_j) - \tilde{a}_{ij},\, \tilde{a}_{ij} - m(\mathbf{x}_i, \mathbf{x}_j)\}. \tag{5.2.15}$$

However, using more than one element can greatly improve the precision of the estimate. If one chooses to use multiple elements of I_{ij}^r, the computations become a bit more complex:

Order the set I_{ij}^r such that

$$\mathbf{x}_k := \{\mathbf{x}_{i,\,k},\, \mathbf{x}_{j,\,k}\} \in I_{ij}^r \qquad \forall\, k \in \mathbb{N}_{\left|I_{ij}^r\right|}. \tag{5.2.16}$$

The overdetermined system generated using I_{ij}^r will be consistent only if

$$\max_k\, m(\mathbf{x}_k) \leq \min_k\, M(\mathbf{x}_k), \tag{5.2.17}$$

so restrict it to the largest subset of I_{ij}^r such that Equation (5.2.17) holds.

Assume without loss of generality that the set I_{ij}^r generates a consistent system, and let

$$m_{ij} := \max_k\, m(\mathbf{x}_k) \qquad \text{and} \qquad M_{ij} := \min_k\, M(\mathbf{x}_k). \tag{5.2.18}$$

The value of $\tilde{a}_{ij}$ takes one of two forms:

Whenever possible, use the average given by

$$\tilde{a}_{ij} := \frac{1}{\left|I_{ij}^r\right|} \sum_{k=1}^{\left|I_{ij}^r\right|} g\left(\frac{\mathbf{x}_{i,\,k} + \mathbf{x}_{j,\,k}}{2}\right). \tag{5.2.19}$$

However, the average must satisfy the relation

$$m_{ij} \leq \tilde{a}_{ij} \leq M_{ij} \tag{5.2.20}$$

is satisfied. If the average does not satisfy Equation (5.2.20), use a fall-back:

$$\tilde{a}_{ij} := \frac{m_{ij} + M_{ij}}{2}. \tag{5.2.21}$$

Regardless of which definition of $\tilde{a}_{ij}$ is applied, the error can be computed as

$$\alpha_{ij} := \max\{M_{ij} - \tilde{a}_{ij},\, \tilde{a}_{ij} - m_{ij}\}. \tag{5.2.22}$$

If $\left|I_{ij}^r\right| = 1$, this method reduces to the one described by Equations (5.2.11) and (5.2.15).

Remark 5.2.5. The set of intersection points is useful in computing the shift differences, but it also serves another purpose. As mentioned in Section 5.2.1, Equation (5.2.5) implies that each set A_{ij} is a subset of some level set of g_{ij}. Hence, the shift difference $\tilde{a}_{ij}$ and the intersection points I_{ij}^r can often be used to write an approximation for A_{ij} as a collection of smooth manifolds (usually curves or surfaces). For example, if A_{ij} is a single smooth manifold, represent it with the implicit equation

$$g_{ij}(\mathbf{x}) = \tilde{a}_{ij} \qquad \text{from } \mathbf{x}_0 \text{ to } \mathbf{x}_1, \tag{5.2.23}$$

where $\mathbf{x}_0$ and $\mathbf{x}_1$ are intersection points approximated using I_{ij}^r.

5.2.2.3 Computing the shifts

Section 5.2.2.2 describes how to choose individual shift differences a_{ij}, but it does not describe *which* shift differences to pick. The number of possible sets of shift differences is equal to the number of unique trees H that span the adjacency graph G. I will consider the appropriate choice here.

As a consequence of Equation (5.2.5), each a_{ij} satisfies $a_{ij} = a_i - a_j$, and therefore

$$a_i = a_j + a_{ij}. \tag{5.2.24}$$

Let $\tilde{a}_i = a_i + \alpha_i$, $\forall i \in \mathbb{N}_n$, define the relation between exact (a_i's) and approximate ($\tilde{a}_i$'s) shifts.

The first shift is assigned a value, and is therefore exact: $\tilde{a}_k = a_k$. Now suppose one uses Equation (5.2.24) to approximate the value of some unknown shift a_i. If a_j was approximated with $\tilde{a}_j = a_j + \alpha_j$, and a_{ij} was approximated with $\tilde{a}_{ij} = a_{ij} + \alpha_{ij}$, then

$$\tilde{a}_i = \tilde{a}_j + \tilde{a}_{ij} = (a_j + \alpha_j) + (a_{ij} + \alpha_{ij}) = a_j + a_{ij} + (\alpha_j + \alpha_{ij}). \tag{5.2.25}$$

Thus, the error in the approximation of a_i could be as large as $|\alpha_j| + |\alpha_{ij}|$.

Inductively, the longer the path in a spanning tree H from v_k (the vertex corresponding to the initial shift) to v_i (the vertex corresponding to the target shift), the more error terms must be taken into account when approximating a_i. For this reason, I recommend computing the shifts using the following greedy approach:

(0) For the initial shift, a_k, choose k such that A_k has the largest number of adjacent regions. [This is equivalent to creating the subgraph $H = v_k$, where v_k is the vertex in G of largest degree.]

While one or more shifts a_i remain unknown:

(1) Choose j such that a_j is known and A_j has the largest number of adjacent regions whose shifts are still unknown. [This is equivalent to choosing the vertex v_j in H with the most adjacent vertices in $G \setminus H$.]

(2) Determine the shift difference a_{ij} for each unknown a_i adjacent to a_j. [This is equivalent to adding the vertices v_i and edges (v_i, v_j) to H.]

(3) For each unknown a_i adjacent to a_j, approximate a_i using Equation (5.2.24).

Once H is the desired spanning tree, all shifts will have been approximated.

5.2.2.4 Approximating the Wasserstein distance

By Step (5), the partial Wasserstein distance $\widetilde{P}$ includes the exact cost of all the identified interior regions. What remains to be done is to approximate the cost of the regions associated with B^r.

Using the transport map obtained during Step (1) of the r-th iteration, for each $\mathbf{x} \in B^r$ one can determine the set

$$\mathbf{y}_\mathbf{x} := \{\mathbf{y} \in Y \mid T(\mathbf{x}_n) = \mathbf{y} \text{ for some } \mathbf{x}_n \in \{\mathbf{x}\} \cup N(\mathbf{x})\}. \tag{5.2.26}$$

The approximated Wasserstein distance over $\mathbf{x}^r$ is given by

$$P_{\mathbf{x}} := \int_{\mathbf{x}^r} c(\mathbf{z},\, T(\mathbf{x}))\, d\mu(\mathbf{z}). \tag{5.2.27}$$

The maximum and minimum possible Wasserstein distance over $\mathbf{x}^r$ equal

$$M_{\mathbf{x}} := \max_{\mathbf{y}\in\mathbf{y}_{\mathbf{x}}} \int_{\mathbf{x}^r} c(\mathbf{z},\, \mathbf{y})\, d\mu(\mathbf{z}) \qquad \text{and} \qquad m_{\mathbf{x}} := \min_{\mathbf{y}\in\mathbf{y}_{\mathbf{x}}} \int_{\mathbf{x}^r} c(\mathbf{z},\, \mathbf{y})\, d\mu(\mathbf{z}) \tag{5.2.28}$$

respectively. Hence, the error bound for the approximated Wasserstein distance over $\mathbf{x}^r$ equals

$$\gamma_{\mathbf{x}} := \max\{M_{\mathbf{x}} - P_{\mathbf{x}},\, P_{\mathbf{x}} - m_{\mathbf{x}}\}. \tag{5.2.29}$$

Summing over all B^r, the approximate Wasserstein distance equals

$$\widetilde{P}^* := \widetilde{P} + \sum_{\mathbf{x}\in B^r} P_{\mathbf{x}}, \tag{5.2.30}$$

and the error bound for the approximate Wasserstein distance is

$$\gamma^* := \sum_{\mathbf{x}\in B^r} \gamma_{\mathbf{x}}. \tag{5.2.31}$$

5.3 Mathematical support

In this section, I provide mathematical justification for the boundary method, assuming that all computations are solved exactly: both the discrete optimal transport problems handled by the auction algorithm, and the determinations of mass and Wasserstein distance for individual boxes (see Remark 5.2.3). I present three types of results: on admissible ground cost functions, on the shift-characterized semi-discrete optimal transport problem, and, finally, results on the boundary method itself.

5.3.1 Ground cost functions

I now define a convenient notation for a large class of ground cost functions, and use that definition to further clarify which ground cost functions are admissible.

Definition 5.3.1 (ℓ_p^q and ℓ_p functions)**.** Let $\overline{\mathbb{R}}^+ := \mathbb{R}^+ \cup \infty$. Suppose there exist $p \in \overline{\mathbb{R}}^+$ and $q > 0$ such that for all $\mathbf{x} = (x_1, \ldots, x_d) \in X$ and $\mathbf{y} = (y_1, \ldots, y_d) \in Y$,

$$c(\mathbf{x}, \mathbf{y}) = \begin{cases} \max\limits_{i \in \mathbb{N}_d} |y_i - x_i|^q & \text{if } p = \infty \\ \left[\sum\limits_{i=1}^{d} |y_i - x_i|^p \right]^{q/p} & \text{otherwise.} \end{cases} \tag{5.3.1}$$

Then the ground cost may be written as $c = \ell_p^q$.

We refer to the set of ground costs

$$\{\, c \mid c = \ell_p^q, \text{ with } p \in \overline{\mathbb{R}}^+ \text{ and } q = 1 \,\} \tag{5.3.2}$$

as the ℓ_p *functions*. One can also write $c = \ell_p$ with $q = 1$ implied.

Theorem 5.3.2. *Classes of admissible ground cost functions:*

(a) If c is an ℓ_p function, then c is an admissible ground cost function, as described in Definition 5.1.2.

(b) If c is an admissible ground cost function and $\tilde{c} = c^q$ for some $q > 0$, then $\tilde{c}$ is also an admissible ground cost function.

(c) If $\{c_i\}_{i=1}^m$ is a set of admissible ground cost functions, and c be a linear combination of $\{c_i\}_{i=1}^m$ with positive coefficients,

$$c := \sum_{i=1}^{m} k_i c_i \qquad \textit{s.t. } k_i > 0 \qquad \forall\, i \in \mathbb{N}_m, \tag{5.3.3}$$

then c is an admissible ground cost function, as described in Definition 5.1.2.

Proof. For Part (a), Properties (1) and (2) of Definition 5.1.2 are self-evident. For Property (3), let $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 \in \mathbb{R}^d$ be collinear points: $\mathbf{x}_1 = \mathbf{x}_2 + a\mathbf{v}$ and $\mathbf{x}_3 = \mathbf{x}_2 + b\mathbf{v}$ for some vector $\mathbf{v} = (v_1, \ldots, v_d)$ and scalars a and b. If $\|\mathbf{x}_1 - \mathbf{x}_2\|_2 \leq \|\mathbf{x}_3 - \mathbf{x}_2\|_2$, then $|a| \leq |b|$. Thus, if $p = \infty$,

$$
\begin{aligned}
|a| \max_{i \in \mathbb{N}_d} |v_i| &\leq |b| \max_{i \in \mathbb{N}_d} |v_i| \\
\max_{i \in \mathbb{N}_d} |av_i| &\leq \max_{i \in \mathbb{N}_d} |bv_i| \\
\max_{i \in \mathbb{N}_d} |x_{2,i} + av_i - x_{2,i}| &\leq \max_{i \in \mathbb{N}_d} |x_{2,i} + bv_i - x_{2,i}| \\
\max_{i \in \mathbb{N}_d} |x_{1,i} - x_{2,i}| &\leq \max_{i \in \mathbb{N}_d} |x_{3,i} - x_{2,i}| \\
c(\mathbf{x}_2, \mathbf{x}_1) &\leq c(\mathbf{x}_2, \mathbf{x}_3).
\end{aligned}
$$

If $p < \infty$, then

$$
\begin{aligned}
|a| \left[\sum_{i=1}^{d} |v_i|^p \right]^{1/p} &\leq |b| \left[\sum_{i=1}^{d} |v_i|^p \right]^{1/p} \\
\left[\sum_{i=1}^{d} |av_i|^p \right]^{1/p} &\leq \left[\sum_{i=1}^{d} |bv_i|^p \right]^{1/p} \\
\left[\sum_{i=1}^{d} |x_{2,i} + av_i - x_{2,i}|^p \right]^{1/p} &\leq \left[\sum_{i=1}^{d} |x_{2,i} + bv_i - x_{2,i}|^p \right]^{1/p} \\
\left[\sum_{i=1}^{d} |x_{1,i} - x_{2,i}|^p \right]^{1/p} &\leq \left[\sum_{i=1}^{d} |x_{3,i} - x_{2,i}|^p \right]^{1/p} \\
c(\mathbf{x}_2, \mathbf{x}_1) &\leq c(\mathbf{x}_2, \mathbf{x}_3).
\end{aligned}
$$

Therefore, all ground cost properties are satisfied.

For Part (b), let $\tilde{c} = c^q$ for some $q > 0$. Then

1. If $\mathbf{x} \in \mathbb{R}^d$, $\tilde{c}(\mathbf{x}, \mathbf{x}) = 0^q = 0$.
2. Let $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$ such that $\mathbf{x}_1 \neq \mathbf{x}_2$. Since $c(\mathbf{x}_1, \mathbf{x}_2) > 0$, $\tilde{c}(\mathbf{x}_1, \mathbf{x}_2) = [c(\mathbf{x}_1, \mathbf{x}_2)]^q > 0$.

3. Let $\mathbf{x}_1,\, \mathbf{x}_2,\, \mathbf{x}_3 \in \mathbb{R}^d$ be collinear points such that $\|\mathbf{x}_1 - \mathbf{x}_2\|_2 \leq \|\mathbf{x}_3 - \mathbf{x}_2\|_2$. Then $c(\mathbf{x}_2,\, \mathbf{x}_1) \leq c(\mathbf{x}_2,\, \mathbf{x}_3)$, and since $q > 0$, $[c(\mathbf{x}_2,\, \mathbf{x}_1)]^q \leq [c(\mathbf{x}_2,\, \mathbf{x}_3)]^q$.

The combination in Part (c) is obvious, in light of Part (b) above. □

5.3.2 Semi-discrete optimal transport and the shift characterization

Here I examine the features of the shift characterization, defined in Section 5.1.2, and consider what they say about the semi-discrete optimal transport problem itself.

First, in Lemmas 5.3.3 and 5.3.4, I develop theoretical support for the boundary method.

Lemma 5.3.3. *Fix $i \in \mathbb{N}_n$. If $\mathbf{x} \in A_i$ and $j \in \mathbb{N}_n$, $j \neq i$, then the following hold:*

$$g_{ij}(\mathbf{x}) \leq a_i - a_j, \tag{5.3.4}$$

$$g_{ij}(\mathbf{x}) = a_i - a_j \qquad \textit{if and only if} \qquad \mathbf{x} \in A_{ij}, \tag{5.3.5}$$

$$g_{ij}(\mathbf{x}) < a_i - a_j \qquad \textit{if and only if} \qquad \mathbf{x} \in A_i \setminus A_j. \tag{5.3.6}$$

Proof. Consider Equation (5.3.4). By the definitions of A_i and F,

$$a_i - c(\mathbf{x},\, y_i) = F(x) \geq a_j - c(\mathbf{x},\, y_j).$$

Rearranging terms gives

$$c(\mathbf{x},\, \mathbf{y}_i) - c(\mathbf{x},\, \mathbf{y}_j) \leq a_i - a_j.$$

To show Equation (5.3.5), first note that Section 5.2.1 already explains how $\mathbf{x} \in A_{ij}$ implies $g_{ij}(\mathbf{x}) = a_i - a_j$. Consider the converse: Assume that $g_{ij}(\mathbf{x}) = a_i - a_j$. Rewriting, one finds that

$$a_j - c(\mathbf{x},\, \mathbf{y}_j) = a_i - c(\mathbf{x},\, \mathbf{y}_i) = F(\mathbf{x}).$$

This implies $\mathbf{x} \in A_j$, and since $\mathbf{x} \in A_i$, therefore $\mathbf{x} \in A_{ij}$.

Equation (5.3.6) is a direct consequence of combining Equations (5.3.4) and (5.3.5).

□

Lemma 5.3.4. *Suppose c satisfies the triangle inequality. For all $i,\ j \in \mathbb{N}_n$, $i \neq j$,*

(a) If $c(\mathbf{y}_i,\ \mathbf{y}_j) = a_i - a_j$, then $A_j \subseteq A_{ij}$.

(b) If $c(\mathbf{y}_i,\ \mathbf{y}_j) < a_i - a_j$, then $A_j = \varnothing$.

Proof. Because c satisfies the triangle inequality, for all $\mathbf{x} \in A$,

$$\begin{aligned} c(\mathbf{x},\ \mathbf{y}_i) &\leq c(\mathbf{x},\ \mathbf{y}_j) + c(\mathbf{y}_i,\ \mathbf{y}_j) \\ c(\mathbf{x},\ \mathbf{y}_i) &\leq c(\mathbf{x},\ \mathbf{y}_j) + a_i - a_j \\ a_j - c(\mathbf{x},\ \mathbf{y}_j) &\leq a_i - c(\mathbf{x},\ \mathbf{y}_i) \end{aligned} \tag{5.3.7}$$

For Part (a), suppose $\mathbf{x} \in A_j$. By Equation (5.3.7), $a_i - c(\mathbf{x},\ \mathbf{y}_i) \geq a_j - c(\mathbf{x},\ \mathbf{y}_j) = F(\mathbf{x})$. Because F is defined as the maximum such difference, this implies $a_i - c(\mathbf{x},\ \mathbf{y}_i) = F(\mathbf{x})$, and so $\mathbf{x} \in A_i$. Further, since $\mathbf{x}$ is an element of A_i and A_j, $\mathbf{x} \in A_{ij}$. Therefore, $A_j \subseteq A_{ij}$.

To show Part (b), note that now Equation (5.3.7) gives $a_j - c(\mathbf{x},\ \mathbf{y}_j) < a_i - c(\mathbf{x},\ \mathbf{y}_i)$. Hence, for all $\mathbf{x} \in A$, $F(\mathbf{x}) \geq a_i - c(\mathbf{x},\ \mathbf{y}_i) > a_j - c(\mathbf{x},\ \mathbf{y}_j)$. Therefore, $A_j = \varnothing$. □

Lemma 5.3.5. *$F(\mathbf{x})$ is a continuous function of $\mathbf{x}$.*

Proof. The ground cost c is defined as a continuous function in $X \times Y$. Thus, for all $i \in \mathbb{N}_n$, $a_i - c(x,\ y_i)$ is a continuous function of $\mathbf{x}$. Since F is the maximum of a finite set of continuous functions, F is itself a continuous function of $\mathbf{x}$. □

The next set of results aim at showing that if F partitions A with respect to the measure μ, then the optimal transport map is unique except on a set of μ-measure zero.

Definition 5.3.6 (F μ-partitions A)**.** Let F be as defined in Equation (5.1.5), and the sets A_i as defined in Equation (5.1.6) for $i \in \mathbb{N}_n$. Then one says F *μ-partitions* the set A, or F is called a *μ-partition*. If

1. $\mu(A) < \infty$,
2. for all $i,\ j \in \mathbb{N}_n$, $i \neq j$, $\mu(A_{ij}) = 0$,

3. $\sum_{i=1}^{n} \mu(A_i) = \mu(A)$, and
4. for all $i \in \mathbb{N}_n$, $\mu(A_i) = \nu(\mathbf{y}_i) > 0$.

Theorem 5.3.7. *Suppose one has a semi-discrete transport problem, as described in Section 5.1.1. Let F be as defined in Equation* (5.1.5)*, and the sets A_i as defined in Equation* (5.1.6) *for $i \in \mathbb{N}_n$. Then F μ-partitions A if and only if $\mu(B) = 0$.*

Proof. If F μ-partitions A, by Definition 5.3.6, $\mu(B) = 0$. For the converse, assume F and the sets A_i are defined as given. Because μ is a probability density function, $\mu(A) = 1 < \infty$. Because μ is a non-negative measure, $\mu(B) = 0$ implies that, for all $i,\ j \in \mathbb{N}_n$, $i \neq j$ $\mu(A_{ij}) = 0$.

For any μ-measurable set $S \subseteq X$, $S = S_1 \cup S_2$,

$$\mu(S_1) + \mu(S_2) = \mu(S) + \mu(S_1 \cap S_2),$$

and since $\mu(X) < \infty$,

$$\mu(S) = \mu(S_1) + \mu(S_2) - \mu(S_1 \cap S_2).$$

Proceeding inductively, it follows that

$$S = \bigcup_{i=1}^{n} S_i, \text{ all } \mu\text{-measurable} \qquad \Longrightarrow \qquad \mu(S) = \sum_{i=1}^{n} \mu(S_i) - \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \mu(S_i \cap S_j).$$

Thus,

$$1 = \mu(A) = \sum_{i=1}^{n} \mu(A_i) - \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \mu(A_{ij}) = \sum_{i=1}^{n} \mu(A_i).$$

For all $i,\ j \in \mathbb{N}_n$, $i \neq j$, $\mu(A_i \cap A_j) = 0$, and therefore $\mu(A_i) = \nu(\mathbf{y}_i)$. □

Definition 5.3.8. A transport plan π is said to be *Monge under the shift characterization* if π is a Monge solution, with an associated transport map T, and there exists a function F, as described in Equation (5.1.5), and sets $\{A_i\}_{i=1}^n$, as described in Equation (5.1.6), such that for all $\mathbf{x} \in A$,

$$\mathbf{x} \in \mathring{A}_i \text{ for some } i \in \mathbb{N}_n \qquad \Longrightarrow \qquad T(\mathbf{x}) = \mathbf{y}_i. \tag{5.3.8}$$

In other words, T agrees with some μ-partition F on $A \setminus B$.

If $\mu(B) > 0$ for the shifts $\{a_i\}_{i=1}^n$, no such transport plan π can exist, and the transport problem itself can be said to be *not Monge under the shift characterization*. Conversely, if $\mu(B) = 0$, then such a transport plan exists, and so the transport problem itself is said to be Monge under the shift characterization.

Corollary 4 of [38] gives a sufficient condition for the existence of a Monge solution that is unique μ-a.e.:

$$\mu\left(\{\, \mathbf{x} \in A \mid g_{ij}(\mathbf{x}) = k \,\}\right) = 0 \qquad \forall\, i,\, j \in \mathbb{N}_n,\, i \neq j, \qquad \forall\, k \in \mathbb{R}. \tag{5.3.9}$$

As I will show, that condition is also sufficient for the a transport problem to be Monge under the shift characterization.

Theorem 5.3.9. *If the semi-discrete transport problem satisfies Equation* (5.3.9)*, then the transport problem is Monge under the shift characterization.*

Proof. Let $i,\, j \in \mathbb{N}_n$, $i \neq j$, and $k = a_i - a_j$. Then $A_{ij} \subseteq \{g_{ij}(\mathbf{x}) = k\}$, which implies

$$\mu(A_{ij}) \leq \mu(\{g_{ij}(\mathbf{x}) = k\}) = 0.$$

Hence, $\mu(B) = 0$, and Theorem 5.3.7 implies that F μ-partitions A. Therefore, by Definition 5.3.8, the semi-discrete transport problem is Monge under the shift characterization. □

If the condition described in Equation (5.3.9) is satisfied, Corollary 4 in [38] implies that a transport map T that solves the semi-discrete problem must be unique except on a set of μ-measure zero. Thus, the condition described in Equation (5.3.9) is sufficient to ensure a solution that is unique μ-a.e. However, the condition is not necessary: for a counter-example, consider the semi-discrete transport problem whose solution is shown in Figure 5.6(a).

As the figure illustrates, $\mu(B) = 0$, and the problem is Monge under the shift characterization. However, the problem does not satisfy the condition given in Equation (5.3.9): Figure 5.6(a) describes a problem in $[0,\, 1]^2$ with $c = \ell_1$, and μ and ν uniform. Assume $\mathbf{y}_3 = (y_{3,1},\, y_{3,2})$. For any $\mathbf{x} = (x_1,\, x_2)$ such that $0 < x_1 \leq y_{3,1}$ and $0 < x_2 \leq y_{3,2}$, $g_{23}(\mathbf{x}) = c(\mathbf{y}_2,\, \mathbf{y}_3)$. Therefore,

$$\mu\left(\{\, \mathbf{x} \in A \mid g_{23}(\mathbf{x}) = c(\mathbf{y}_2,\, \mathbf{y}_3)\,\}\right) \geq x_1 x_2 > 0.$$

Because Equation (5.3.9) is not a necessary condition for a Monge solution to be unique μ-a.e., we offer the following, more general, conditions under which the optimal transport map is unique μ-a.e.:

Theorem 5.3.10 (The optimal transport map is unique μ-a.e.)**.** *Given a semi-discrete transport problem, let π^* and $\tilde{\pi}^*$ be optimal transport plans that are both Monge under the shift characterization. If T is a transport map associated with π^*, and $\widetilde{T}$ a transport map associated with $\tilde{\pi}^*$, then $T = \widetilde{T}$ except on a set of μ-measure zero.*

Proof. Assume to the contrary that T and $\widetilde{T}$ differ on a set of μ-positive measure. I will show that the difference between the Wasserstein distances $P(\widetilde{\pi}^*)$ and $P(\pi^*)$ must be non-zero, thereby establishing the desired contradiction: at least one of π^* and $\widetilde{\pi}^*$ is not optimal.

Because π^* is Monge under the shift characterization, there exist F and $\{A_i\}_{i=1}^n$ that satisfy the definitions in Equations (5.1.5) and (5.1.6). Similarly, one has $\widetilde{F}$ and $\{\tilde{A}_i\}_{i=1}^n$ for $\widetilde{\pi}^*$. Let $\{a_i\}_{i=1}^n$ be the set of shifts associated with F, and $\{\tilde{a}_i\}_{i=1}^n$ the shifts associated

with $\widetilde{F}$. For all $i,\ j \in \mathbb{N}_n$, $i \neq j$, define

$$X_{ij} := \Big\{ \mathbf{x} \in A \mid T(\mathbf{x}) = \mathbf{y}_i \quad \text{and} \quad \widetilde{T}(\mathbf{x}) = \mathbf{y}_j \Big\}.$$

Since F μ-partitions A, $\mu(A_{ij}) = 0$. The Wasserstein distance is only affected by sets of positive measure, so without loss of generality assume $X_{ij} \cap A_{ij} = \varnothing$.

With respect to the transformation from π^* to the map $\tilde{\pi}^*$, the change in Wasserstein distance over the set X_{ij} equals

$$\begin{aligned} \Delta P_{ij} &:= \int_{X_{ij}} c(\mathbf{x},\, \widetilde{T}(\mathbf{x}))\, d\mu(\mathbf{x}) - \int_{X_{ij}} c(\mathbf{x},\, T(\mathbf{x}))\, d\mu(\mathbf{x}) \\ &= \int_{X_{ij}} \left[c(\mathbf{x},\, \mathbf{y}_j) - c(\mathbf{x},\, \mathbf{y}_i) \right]\, d\mu(\mathbf{x}). \end{aligned}$$

Since $X_{ij} \cap A_{ij} = \varnothing$, $\mathbf{x} \in X_{ij}$ implies $\mathbf{x} \in A_i \setminus A_j$. Hence, by Equation (5.3.6),

$$\forall \mathbf{x} \in X_{ij}, \qquad g_{ij}(\mathbf{x}) < a_i - a_j.$$

Therefore, for all $i,\ j \in \mathbb{N}_n$, $i \neq j$ implies $\Delta P_{ij} \geq (a_j - a_i)\mu(X_{ij})$, with equality if and only if $\mu(X_{ij}) = 0$.

Let

$$X_i = \bigcup_{\substack{j=1 \\ j \neq i}}^{n} X_{ij} \qquad \text{and} \qquad \Delta P_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} \Delta P_{ij}.$$

If i, j, and k are pairwise distinct, by definition $X_{ij} \cap X_{ik} = \varnothing$. Therefore

$$\mu(X_i) = \sum_{\substack{j=1 \\ j \neq i}}^{n} \mu(X_{ij}) \qquad \text{and} \qquad \Delta P_i \geq \sum_{\substack{j=1 \\ j \neq i}}^{n} (a_j - a_i)\mu(X_{ij}).$$

By the definition of the shift characterization, $\mu(A_i) = \mu(\tilde{A}_i)$. Thus,

$$\mu(A_i) = \mu(\tilde{A}_i) = \mu\left((A_i \setminus X_i) \cup \left(\bigcup_{\substack{j=1 \\ j\neq i}}^{n} X_{ji}\right)\right),$$

and because the sets are disjoint, this implies

$$\mu(A_i) = \mu(A_i) - \mu(X_i) + \sum_{\substack{j=1 \\ j\neq i}}^{n} \mu(X_{ji}), \qquad \text{and therefore} \qquad \mu(X_i) = \sum_{\substack{j=1 \\ j\neq i}}^{n} \mu(X_{ji}).$$

Since T and $\widetilde{T}$ differ on a set of positive measure, there exists at least one pair $i,\ j \in \mathbb{N}_n$ such that $i \neq j$ and $\mu(X_{ij}) > 0$. For that i and j, $\Delta P_{ij} > (a_j - a_i)\mu(X_{ij})$, which implies

$$\Delta P_i > \sum_{\substack{j=1 \\ j\neq i}}^{n} (a_j - a_i)\mu(X_{ij}).$$

Thus,

$$\Delta P := \sum_{i=1}^{n} \Delta P_i > \sum_{i=1}^{n} \left(\sum_{\substack{j=1 \\ j\neq i}}^{n} (a_j - a_i)\mu(X_{ij}) \right).$$

By rearranging terms and indexing appropriately

$$\begin{aligned}
\sum_{i=1}^{n} \left(\sum_{\substack{j=1 \\ j\neq i}}^{n} (a_j - a_i)\mu(X_{ij}) \right) &= \left(\sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq i}}^{n} a_j \mu(X_{ij}) \right) - \sum_{i=1}^{n} a_i \left(\sum_{\substack{j=1 \\ j\neq i}}^{n} \mu(X_{ij}) \right) \\
&= \left(\sum_{j=1}^{n} \sum_{\substack{i=1 \\ j\neq i}}^{n} a_i \mu(X_{ji}) \right) - \sum_{i=1}^{n} a_i \mu(X_i) = \left(\sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq i}}^{n} a_i \mu(X_{ji}) \right) - \sum_{i=1}^{n} a_i \mu(X_i) \\
&= \sum_{i=1}^{n} a_i \left(\sum_{\substack{j=1 \\ j\neq i}}^{n} \mu(X_{ji}) \right) - \sum_{i=1}^{n} a_i \mu(X_i) = \sum_{i=1}^{n} a_i \left(-\mu(X_i) + \sum_{\substack{j=1 \\ j\neq i}}^{n} \mu(X_{ji}) \right)
\end{aligned}$$

$$= \sum_{i=1}^{n} a_i(0) = 0.$$

Thus, $\Delta P > 0$, which implies $P(\tilde{\pi}^*) > P(\pi^*)$. This contradicts the definition of $\tilde{\pi}^*$ as optimal. Therefore $\widetilde{T}$ and T must differ only on sets of μ-measure zero. □

Remark 5.3.11 (When F does not μ-partition A)**.** For some problems, the semi-discrete problem will not be Monge under the shift characterization. In other words, there exists some $i, j \in \mathbb{N}_n$, $i \neq j$, such that $\mu(A_{ij}) > 0$. Unfortunately, this important fact has not always been recognized or clearly expressed in the literature [102, 103, 104]. The resulting confusion has been further aggravated by a lack of specific examples.

Even when a problem is not Monge under the shift characterization, the sets $\{A_i\}_{i=1}^n$ generated by the shifts still form a valid solution. $\mathbf{x} \in A_{ij}$ merely indicates that $\mathbf{x}$ is transported to both $\mathbf{y}_i$ and $\mathbf{y}_j$. When $\mu(A_{ij}) = 0$, one can safely disregard this splitting of mass and choose to send $\mathbf{x}$ to one of the two points $\mathbf{y}_i$ or $\mathbf{y}_j$. In this way, a semi-discrete transport problem that is Monge under the shift characterization guarantees a Monge solution which is unique μ-a.e.

However, when $\mu(A_{ij}) > 0$, some of the points $\mathbf{x} \in A_{ij}$ must have their mass split in order to satisfy ν. Thus, the solution generated by the shift characterization is not Monge. This necessitates an extra step: determining an appropriate mass splitting. The split for the undetermined mass can be solved as a linear programming problem. The constants and unknowns necessary for this linear program are described in Remark 5.3.12.

It is worth pointing out that a Monge solution is still guaranteed to exist, even though the shift characterization is not itself Monge. For this reason, solving the linear program is not necessary when applying the boundary method.

The boundary method identifies the appropriate bounds for the sets $\{A_i\}_{i=1}^n$, but it does so while seeking a Monge solution to the semi-discrete problem. For this problem, at least one Monge solutions is guaranteed to exist. Theorem 5.3.10 does not apply, so the

boundary method's Monge solution is not guaranteed to be unique μ-a.e. However, finding any such solution would be sufficient to determine the Wasserstein distance.

Given these complications, my theoretical results focus on the case where the semi-discrete problem is known to be Monge under the shift characterization. In practice, even if $\mu(B) > 0$, and the solution is neither Monge nor unique, there is no impact whatsoever on the boundary method. Whether or not the shift characterization results in a partition, the boundary method appears to have no difficulty solving transport problems. The presence or absence of a partition also seems irrelevant to the rate of convergence for the Wasserstein distance approximation. See Section 5.5.3.1 for an example.

Remark 5.3.12 (Distribution of mass splitting)**.** For all $\omega \subseteq \mathbb{N}_n$, define

$$A_\omega := \left\{ \mathbf{x} \in \bigcap_{i \in \omega} A_i \;\middle|\; j \notin \omega \implies \mathbf{x} \notin A_j \right\}, \tag{5.3.10}$$

and let

$$\mathcal{W} := \{\, \omega \subseteq \mathbb{N}_n \mid \mu(A_\omega) > 0 \,\}. \tag{5.3.11}$$

Assume $\mathcal{W}$ is ordered:

$$\mathcal{W} = \{\omega_1, \omega_2, \ldots, \omega_K\}. \tag{5.3.12}$$

Then

$$\sum_{i=1}^{K} \mu(A_{\omega_i}) = \mu(A). \tag{5.3.13}$$

For each $i \in \mathbb{N}_n$ and $j \in \omega_i$, there exist constants $\beta_{i,\,j} \in [0,\, 1]$ such that

$$\sum_{j \in \omega_i} \beta_{i,\,j} = 1 \qquad \forall\, i \in \mathbb{N}_n \tag{5.3.14}$$

and

$$\sum_{\{\, i \in \mathbb{N}_n \mid j \in \omega_i \,\}} \beta_{i,\,j} \mu(A_{\omega_i}) = \nu(\mathbf{y}_j) \qquad \forall\, j \in \mathbb{N}_n. \tag{5.3.15}$$

Furthermore, these constants satisfy

$$\sum_{i=1}^{K} \sum_{j \in \omega_i} \beta_{i,\,j} \mu(A_{\omega_i}) = \mu(A). \tag{5.3.16}$$

Suppose that

$$\mathcal{W} = \{\{1\},\, \{2\},\, \ldots,\, \{n\}\}. \tag{5.3.17}$$

Then for all $i,\, j \in \mathbb{N}_n$,

$$\beta_{i,\,j} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise,} \end{cases} \tag{5.3.18}$$

and therefore

$$\sum_{i=1}^{n} \mu(A_i) = \mu(A). \tag{5.3.19}$$

This is the partitioning described in [102, 104],

$$\mu(A_i) = \nu(\mathbf{y}_i) \qquad \forall\, i \in \mathbb{N}_n, \tag{5.3.20}$$

and it occurs if and only if $\mu(B) = 0$.

5.3.3 Existence of $(n-1)$ functionally independent boundary equations

To prove the existence of $(n-1)$ functionally independent equations of the form shown in Equation (5.2.6), I will investigate the structure of their intersections using a connected graph G defined as follows:

Definition 5.3.13. Let G be a graph with n vertices $v_1,\, \ldots,\, v_n$. The edge $(v_i,\, v_j)$ is contained in the edge set of G if and only if A_{ij} is non-empty. Refer to G as the *adjacency graph* of the transport problem.

Theorem 5.3.14. *G is a strongly connected graph.*

Proof. First, note that G is an undirected graph. Hence, G is strongly connected if and only if it is connected.

Now assume to the contrary that G is not a connected graph. Then one can write G as the union of two disjoint subgraphs, $G = G_1 \cup G_2$, such that no vertex v_1 in the vertex set of G_1 has a path connecting it to any vertex v_2 in G_2.

Define

$$A_1 := \bigcup_{v_i \in G_1} A_i \qquad \text{and} \qquad A_2 := \bigcup_{v_j \in G_2} A_j.$$

Because G_1 and G_2 are disjoint, and no paths connect them, it follows that $A_1 \cap A_2 = \varnothing$. Since the union of G_1 and G_2 is G, $A_1 \cup A_2 = A$.

Suppose $A_i \subseteq A_1$, $A_j \subseteq A_2$. Then $A_{ij} = \varnothing$. A is a closed and bounded set, and the definition given in Equation (5.1.6) implies that A_i and A_j must also be closed and bounded. Thus, A_i and A_j are disjoint compact sets in the Hausdorff space $\mathbb{R}^d$. This implies A_i and A_j are separated by some positive distance ϵ_{ij}. Because this is true for all $A_i \subseteq A_1$ and $A_j \subseteq A_2$, there exists $\epsilon > 0$, the minimum positive distance over all such ϵ_{ij}.

Let $\mathbf{x}_1 \in A_1$, $\mathbf{x}_2 \in A_2$, and for all $t \in [0,\, 1]$, define

$$\mathbf{x}_t = (1 - t)\mathbf{x}_1 + t\mathbf{x}_2.$$

Because $\epsilon > 0$, there exists $(t_0,\, t_1) \subseteq [0,\, 1]$, $|t_1 - t_0| \geq \epsilon$, such that $t \in (t_0,\, t_1)$ implies $\mathbf{x}_t \notin A_1 \cup A_2 = A$. This contradicts the convexity of A.

Hence, G is connected, which implies G is strongly connected. □

Corollary 5.3.15. *For all $i \in \mathbb{N}_n$, there exists some $j \in \mathbb{N}_n$, such that $j \neq i$ and $A_{ij} \neq \varnothing$.*

Proof. Suppose to the contrary that i exists such that for all $j \in \mathbb{N}_n$ such that $j \neq i$, $A_{ij} = \varnothing$. Since $n \geq 2$, G includes at least two vertices, and v_i is disconnected from the rest of G, which contradicts Theorem 5.3.14 □

Corollary 5.3.16. *The boundary set B is nonempty, and for each $\mathbf{x} \in B$, there exist $i,\ j \in \mathbb{N}_n$ such that $i \neq j$ and $\mathbf{x} \in A_{ij}$.*

Proof. This follows directly from Corollary 5.3.15 and the definition of B given in Equation (5.2.2). □

Theorem 5.3.17. *Let G be the adjacency graph of the transport problem, as defined in Definition 5.3.13, and let H be a subgraph of G that includes all n vertices. Define the system of equations*

$$S := \{a_i - a_j = a_{ij} \mid (v_i,\, v_j) \in \textit{ the edge set of } H\}. \tag{5.3.21}$$

The system of equations S is functionally independent with respect to the shifts $\{a_i\}_{i=1}^n$ if and only if H contains no cycles.

Proof. We prove the contrapositive: H contains a cycle if and only if the system of equations S is functionally dependent.

($\Longrightarrow$) Suppose H contains the cycle $(v_{i_1},\, v_{i_2},\, \ldots,\, v_{i_k},\, v_{i_1})$. Then S contains the linear system

$$M \begin{bmatrix} a_{i_1} \\ a_{i_2} \\ \vdots \\ a_{i_{k-1}} \\ a_{i_k} \end{bmatrix} = \begin{bmatrix} a_{i_1 i_2} \\ a_{i_2 i_3} \\ \vdots \\ a_{i_{k-1} i_k} \\ a_{i_k i_1} \end{bmatrix}, \qquad \text{where } M = \begin{bmatrix} 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & & \ddots & & \\ & & & & 1 & -1 \\ -1 & & & & & 1 \end{bmatrix}.$$

Because $\det(M) = 0$, S is functionally dependent.

($\Longleftarrow$) Suppose instead that S is functionally dependent. Given the form of the equations in S, one can assume without loss of generality that S is contains the equations

$$a_{i_j i_{j+1}} = a_{i_j} - a_{i_{j+1}} \qquad \forall j \in \mathbb{N}_{k-1},$$

and that $a_{i_1 i_k} = a_{i_1} - a_{i_k}$ is also in S. By the definition of S, these equations imply that the edges (v_1, v_2), (v_2, v_3), $\ldots$, (v_{k-1}, v_k), and (v_k, v_1) are contained in H. Together, these edges generate the cycle $(v_{i_1}, v_{i_2}, \ldots, v_{i_k}, v_{i_1})$, so H contains at least one cycle. □

Theorem 5.3.18. *There exists at least one system of exactly $(n-1)$ equations of the form $a_i - a_j = a_{ij}$ that is functionally independent with respect to the set of shifts $\{a_i\}_{i=1}^n$. No system of n independent such equations exists.*

Proof. Because G is a connected graph, one can always create a spanning tree H that is a subgraph of G. Let S be the corresponding set of linear equations, defined as described in Equation (5.3.21). As a spanning tree, H contains $(n-1)$ edges and H has no cycles, so by Theorem 5.3.17, S contains exactly $(n-1)$ functionally independent equations.

Suppose a set S of n functionally independent equations exists, all of the form $a_i - a_j = a_{ij}$. Because there are n unknowns in the set of shifts, there is at most one solution set $\{a_i\}_{i=1}^n$. Fix $\sigma \neq 0$ and for all $i \in \mathbb{N}_n$, define $\tilde{a}_i = a_i + \sigma$. For each equation in S, $\tilde{a}_i - \tilde{a}_j = a_i - a_j = a_{ij}$. Thus, $\{\tilde{a}_i\}_{i=1}^n$ is also a solution to S. This contradicts the uniqueness of $\{a_i\}_{i=1}^n$, and therefore no such set of n functionally independent equations exists. □

5.3.4 Discretization for the boundary method

In the first two subsections below, I give some results on how the grid-points interact with the underlying space. Sections 5.3.4.3 and 5.3.4.4 present error bounds. Section 5.3.4.5 considers issues of volume and containment: here I ensure that one can have $B \subseteq \bar{B}^r$ for all r, and show that $|\bar{B}^r| \to 0$ as $r \to \infty$. Finally, Section 5.3.4.6 puts bounds on the error for the Wasserstein distance approximation.

5.3.4.1 Discretization

As described in Section 5.2.2, I discretize the region A using a regular Cartesian grid, and refine the grid over multiple iterations, with the aim of refining only the grid region containing the boundary set.

Let $r \in \mathbb{N}$ be the current discretization level, and $w = w_r$ be the *width* of the discretization at level r. Let A^r be the r-th *point set*, the set of points $\mathbf{x}$ included in the r-th discretization of A. Since one discards points of μ-measure zero during the transport step, assume without loss of generality that $\mu(\mathbf{x}^r) > 0$ for all $\mathbf{x} \in A^r$. Let $A_i^r = A_i \cap A^r$ for $i \in \mathbb{N}_n$.

Let $\mathcal{V}$ be the set of adjacency vectors for all discretizations of A. The adjacency vectors must satisfy the following essential properties:

1. If $\mathbf{v} = (v_1, \ldots, v_d) \in \mathcal{V}$, then $|v_i| \leq 1$ for all $i \in \{1, \ldots, d\}$.
2. $\mathbf{0} \notin \mathcal{V}$.
3. $\mathbf{v} \in \mathcal{V}$ implies $-\mathbf{v} \in \mathcal{V}$.

Because the Cartesian grid offers several computational advantages, for this exposition I choose $\mathcal{V}$ to be the set of linear combinations of the standard unit vectors, $e_1, \ldots, e_d$, with coefficients restricted to ± 1. To satisfy Property (2), I specifically exclude the zero vector from the set, and so $|\mathcal{V}| = 3^d - 1$. If $d = 2$, $\mathcal{V}$ equals

$$\mathcal{V} := \{\, (-1, -1),\, (0, -1),\, (-1, 0),\, (-1, 1),\, (1, -1),\, (1, 0),\, (0, 1),\, (1, 1) \,\}. \quad (5.3.22)$$

The properties of the adjacency vectors are required in order to assure that, for all r and all $\mathbf{x} \in A^r$, the points in A^r that are adjacent to $\mathbf{x}$ constitute a subset of the *neighbors* of $\mathbf{x}$,

$$N(\mathbf{x}) := \{\mathbf{x} + w_r \mathbf{v} \mid \mathbf{v} \in \mathcal{V}\}. \quad (5.3.23)$$

Lemma 5.3.19. *If* $\mathbf{x} \in N(\mathbf{x}_0)$*, then* $\mathbf{x}_0 \in N(\mathbf{x})$.

Proof. Fix $\mathbf{x}_0 \in A^r$, and assume $\mathbf{x} \in N(\mathbf{x}_0)$. Then there exists $\mathbf{v} \in \mathcal{V}$ such that $\mathbf{x} = \mathbf{x}_0 + w_r\mathbf{v}$. Rewriting the equation, $\mathbf{x}_0 = \mathbf{x} + w_r(-\mathbf{v})$. By Property (3) of the adjacency vector set, $-\mathbf{v} \in \mathcal{V}$. Therefore, by the definition of N in Equation (5.3.23), $\mathbf{x}_0 \in N(\mathbf{x})$. $\square$

I now formalize the idea of the r-th interior and boundary point sets used in the discretization. For all $i \in \mathbb{N}_n$, define the r-th iteration *interior point set* associated with A_i as

$$\mathring{A}_i^r := \{\mathbf{x} \in A_i^r \mid \forall \mathbf{v} \in \mathcal{V},\ \mathbf{x} + w_r\mathbf{v} \in A_j^r \implies j = i\}. \tag{5.3.24}$$

Define the r-th *boundary point set* as

$$B^r = A^r \setminus \bigcup_{i=1}^{n} \mathring{A}_i^r, \tag{5.3.25}$$

and let $B_i^r = B^r \cap A_i$ for all $i \in \mathbb{N}_n$. The r-th *evaluation region*, the subset of A enclosed by the discretization A^r, is defined as

$$\bar{A}^r := \{\, \mathbf{x}^r \mid \mathbf{x} \in A^r \,\}, \tag{5.3.26}$$

and the r-th *boundary region*, the subset of A enclosed by the boundary point set B^r, is given by

$$\bar{B}^r := \{\, \mathbf{x}^r \mid \mathbf{x} \in B^r \,\}. \tag{5.3.27}$$

5.3.4.2 Distance bounds

Though the discretization is fully defined, it still needs to be related back to the sets A_{ij} and the boundary set B. To do this, I first bound the maximum possible distance separating A_{ij} and the points in $\bar{B}^r$.

Theorem 5.3.20. *Suppose $B_i^r \neq \varnothing$. For each $\mathbf{x}_i \in B_i^r$, there exists a point $\mathbf{x}_j = \mathbf{x}_i + w_r\mathbf{v}$, with $\mathbf{v} \in \mathcal{V}$, such that $\mathbf{x}_j \in B_j^r$ for some $j \neq i$. The distance from $\mathbf{x}_i$ to the set A_{ij}, as measured with respect to the ground cost c, is bounded above by $c(\mathbf{x}_i, \mathbf{x}_j)$.*

Proof. Because $\mathbf{x}_i \in B_i^r$, by the definition of B^r, there exists $\mathbf{x}_j = \mathbf{x}_i + w_r\mathbf{v} \in A_j^r \cup N(\mathbf{x}_0)$ for some $j \neq i$. By Lemma 5.3.19, $\mathbf{x}_i \in N(\mathbf{x}_j)$, and since $\mathbf{x}_i \in A_i^r$, it follows that $\mathbf{x}_j \in B_j^r$. Thus, $\mathbf{x}_i \in A$ and $\mathbf{x}_j \in A$, and because A is convex, this implies

$$\{t\mathbf{x}_i + (1-t)\mathbf{x}_j \mid t \in [0,\, 1]\} \subseteq A.$$

By Lemma 5.3.5, F is continuous on A. Therefore, because $\mathbf{x}_i \in A_i$ and $\mathbf{x}_j \in A_j$, there exists $t_* \in [0,\, 1]$ such that $\mathbf{b} = t_*\mathbf{x}_i + (1-t_*)\mathbf{x}_j \in A_{ij}$. Note that $\mathbf{b} = \mathbf{x}_i + (1-t_*)w_r\mathbf{v}$, so all three points are collinear. Applying the ℓ_2 norm,

$$\|\mathbf{b} - \mathbf{x}_i\|_2 = \|(1-t_*)w_r\mathbf{v}\|_2 = (1-t_*)\|w_r\mathbf{v}\|_2 \leq \|w_r\mathbf{v}\|_2 = \|\mathbf{x}_j - \mathbf{x}_i\|_2.$$

Since c is an admissible cost function and $\|\mathbf{b} - \mathbf{x}_i\|_2 \leq \|\mathbf{x}_j - \mathbf{x}_i\|_2$, this implies $c(\mathbf{x}_i,\, \mathbf{b}) \leq c(\mathbf{x}_i,\, \mathbf{x}_j)$. □

Because one can bound the ground cost between the points in B^r and the set A_{ij} in terms of the ground cost between neighboring points, it is worth identifying a bound on that ground cost between neighbors.

First, I ensure that some bound exists for every ground cost function and iteration r.

Lemma 5.3.21. *For all $\mathbf{x}_i \in B_i^r$ and $\mathbf{x}_j \in B_j^r$, where $i,\, j \in \mathbb{N}_n$ and $i \neq j$, if $\mathbf{x}_j \in N(\mathbf{x}_i)$, then $c(\mathbf{x}_i,\, \mathbf{x}_j) < M_r$ for some bounded value M_r.*

Proof. Let $\mathbf{x}_i$ and $\mathbf{x}_j$ be defined as above. By applying the definition given in Equation (5.3.25), $\mathbf{x}_j = \mathbf{x}_i + w_r\mathbf{v}$ for some $\mathbf{v} \in \mathcal{V}$. Further, one can assume without loss of generality that $c(\mathbf{x}_i,\, \mathbf{x}_j)$ is maximized among the finite set of neighbors of $\mathbf{x}_i$. Let M_r be defined as the maximum of all such ground costs over the set of points in B^r:

$$M_r = \max_{\substack{\mathbf{x}_i \in B_i^r \\ 1 \leq i \leq n}} \; \max_{\{\mathbf{x}_j = \mathbf{x}_i + w_r\mathbf{v} \in B_j^r \mid j \neq i\}} c(\mathbf{x}_i,\, \mathbf{x}_j). \tag{5.3.28}$$

Because M_r is the maximum of a finite set of finite values, $M_r < \infty$. □

I can give a specific ground cost bound in terms of w_r for neighbors $\mathbf{x}_0$ and $\mathbf{x}_1$ when $\mathcal{V}$ is the Cartesian grid and c is a polynomial combination of ℓ_p functions with positive coefficients.

Corollary 5.3.22. *Suppose $\mathcal{V}$ defines the Cartesian grid on $\mathbb{R}^d$, as described above, and c is a polynomial combination of ℓ_p functions with positive coefficients:*

$$c = \sum_{s=1}^{\sigma} k_s \ell_{p_s}^{q_s} \text{ such that } k_s,\ p_s,\ q_s > 0 \text{ for all } s \in \mathbb{N}_\sigma. \tag{5.3.29}$$

Let $q = \min_{1 \le s \le \sigma} q_s$. Then the maximum M_r, as described in Lemma 5.3.21, satisfies

$$M_r \le (w_r)^q \sum_{s=1}^{\sigma} k_s (w_r)^{q_s - q} (d)^{q_s/p_s}. \tag{5.3.30}$$

Proof. Let $\mathbf{x} \in B^r$ and fix $s \in \mathbb{N}_\sigma$. For all $\mathbf{v} \in \mathcal{V}$, the subcost $c_s = \ell_{p_s}^{q_s}$ can be computed as

$$c_s(\mathbf{x},\ \mathbf{x} + w_r\mathbf{v}) = \|(\mathbf{x} + w_r\mathbf{v}) - \mathbf{x}\|_{p_s}^{q_s} = w_r^{q_s} \|\mathbf{v}\|_{p_s}^{q_s}.$$

On the Cartesian grid $\mathcal{V}$, $\|\mathbf{v}\|_{p_s}^{q_s}$ achieves its maximum when $\mathbf{v} = \mathbb{1}_d$, so

$$w_r^{q_s} \|\mathbf{v}\|_{p_s}^{q_s} \le w_r^{q_s} \|\mathbb{1}_d\|_{p_s}^{q_s} = \left(w_r d^{1/p_s}\right)^{q_s}.$$

Thus,

$$c(\mathbf{x},\ \mathbf{x} + w_r\mathbf{v}) = \sum_{s=1}^{\sigma} k_s c_s(\mathbf{x},\ \mathbf{x} + w_r\mathbf{v}) \le \sum_{s=1}^{\sigma} k_s \left(w_r d^{1/p_s}\right)^{q_s}.$$

By determining $q = \min_{1 \le s \le \sigma} q_s > 0$, and factoring out w_r^q, one arrives at Equation (5.3.30).

□

5.3.4.3 *Error bounds for shift differences*

In order to bound the error on the Wasserstein distance, I merely require a finite bound on the errors for the individual shift differences, a_{ij}. However, accurately computing the shift differences themselves is also important, and for that reason, I also present theorems that more finely bound the error on a_{ij} for important ground cost functions. Because estimates are generated using one or more computations of $g_{ij}(\mathbf{x})$, the magnitude of these errors is dependent on the point(s) chosen.

Theorem 5.3.23. *Let $\mathbf{x} \in A$ and $i, j \in \mathbb{N}_n$ such that $i \neq j$. When estimating the shift difference a_{ij} using $\mathbf{x}$, the estimation error equals $|\alpha_{ij}(\mathbf{x})|$, where*

$$\alpha_{ij}(\mathbf{x}) := [c(\mathbf{x}, \mathbf{y}_i) - c(\mathbf{b}, \mathbf{y}_i)] + [c(\mathbf{b}, \mathbf{y}_j) - c(\mathbf{x}, \mathbf{y}_j)], \tag{5.3.31}$$

and $\mathbf{b}$ is the point in A_{ij} nearest to $\mathbf{x}$ with respect to the ground cost.

Proof. Assume $\mathbf{b} \in A_{ij}$ is the closest point in A_{ij} to $\mathbf{x}$. Then

$$c(\mathbf{b}, \mathbf{y}_i) - c(\mathbf{b}, \mathbf{y}_j) = g_{ij}(\mathbf{b}) = a_{ij}.$$

For every $\mathbf{x} \in A$, there exists some $\alpha_{ij}(\mathbf{x}) \in \mathbb{R}$ such that

$$c(\mathbf{x}, \mathbf{y}_i) - c(\mathbf{x}, \mathbf{y}_j) = a_{ij} + \alpha_{ij}.$$

By rearrangement and substitution,

$$\begin{aligned}
\alpha_{ij}(\mathbf{x}) &= -a_{ij} + c(\mathbf{x}, \mathbf{y}_i) - c(\mathbf{x}, \mathbf{y}_j) \\
&= -[c(\mathbf{b}, \mathbf{y}_i) - c(\mathbf{b}, \mathbf{y}_j)] + c(\mathbf{x}, \mathbf{y}_i) - c(\mathbf{x}, \mathbf{y}_j) \\
&= [c(\mathbf{x}, \mathbf{y}_i) - c(\mathbf{b}, \mathbf{y}_i)] + [c(\mathbf{b}, \mathbf{y}_j) - c(\mathbf{x}, \mathbf{y}_j)].
\end{aligned}$$

□

Theorem 5.3.24. *Suppose $\mathcal{V}$ is the Cartesian grid and c is a polynomial combination of ℓ_p functions with positive coefficients, as defined in Equation* (5.3.29)*. For any $\mathbf{x} \in A$ and $i, j \in \mathbb{N}_n$ such that $i \neq j$, $|\alpha_{ij}(\mathbf{x})| < \infty$.*

Proof. Because all costs are nonnegative, and achieve their maximum with $c(\mathbf{0}, l\mathbb{1}_d)$,

$$\begin{aligned}|\alpha_{ij}(\mathbf{x})| &= |[c(\mathbf{x}, \mathbf{y}_i) - c(\mathbf{b}, \mathbf{y}_i)] + [c(\mathbf{b}, \mathbf{y}_j) - c(\mathbf{x}, \mathbf{y}_j)]| \\ &\leq |c(\mathbf{0}, l\mathbb{1}_d) - 0| + |c(\mathbf{0}, l\mathbb{1}_d) - 0| = 2c(\mathbf{0}, l\mathbb{1}_d) = 2\sum_{s=1}^{\sigma} k_s \left(ld^{1/p_s}\right)^{q_s} < \infty. \square\end{aligned}$$

Theorem 5.3.25. *Assume that, in addition to satisfying the ground cost properties, c satisfies the triangle inequality. If a_{ij} is estimated using $\mathbf{x} \in A$, then $|\alpha_{ij}(\mathbf{x})| \leq 2c(\mathbf{x}, \mathbf{b})$, where $\mathbf{b}$ is the point in A_{ij} nearest to x with respect to the ground cost.*

Proof. Applying Equation (5.3.31),

$$\begin{aligned}|\alpha_{ij}(\mathbf{x})| &= |[c(\mathbf{x}, \mathbf{y}_i) - c(\mathbf{b}, \mathbf{y}_i)] + [c(\mathbf{b}, \mathbf{y}_j) - c(\mathbf{x}, \mathbf{y}_j)]| \\ &\leq |c(\mathbf{x}, \mathbf{y}_i) - c(\mathbf{b}, \mathbf{y}_i)| + |c(\mathbf{b}, \mathbf{y}_j) - c(\mathbf{x}, \mathbf{y}_j)|.\end{aligned}$$

Since c satisfies the triangle inequality,

$$c(\mathbf{x}, \mathbf{y}_i) - c(\mathbf{b}, \mathbf{y}_i) \leq c(\mathbf{x}, \mathbf{b}) + c(\mathbf{b}, \mathbf{y}_i) - c(\mathbf{b}, \mathbf{y}_i) = c(\mathbf{x}, \mathbf{b}).$$

Thus,

$$|c(\mathbf{x}, \mathbf{y}_i) - c(\mathbf{b}, \mathbf{y}_i)| \leq |c(\mathbf{x}, \mathbf{b})| = c(\mathbf{x}, \mathbf{b}),$$

and, by a similar line of reasoning, $|c(\mathbf{b}, \mathbf{y}_j) - c(\mathbf{x}, \mathbf{y}_j)| \leq c(\mathbf{x}, \mathbf{b})$. Therefore,

$$|\alpha_{ij}(\mathbf{x})| \leq |c(\mathbf{x}, \mathbf{y}_i) - c(\mathbf{b}, \mathbf{y}_i)| + |c(\mathbf{b}, \mathbf{y}_j) - c(\mathbf{x}, \mathbf{y}_j)| \leq 2c(\mathbf{x}, \mathbf{b}). \qquad \square$$

In addition to bounding the error for individual points $\mathbf{x}$, I can also establish meaningful global bounds.

Definition 5.3.26. Let $\alpha_{\max}$ be the maximum value of $|\alpha_{ij}(\mathbf{x})|$ over all $\mathbf{x} \in \bar{B}^r$ and $i,\ j \in \mathbb{N}_n$, such that: (1) $i \neq j$, (2) $\mathbf{x} \in \mathbf{x}_i^r$ for some $\mathbf{x}_i \in B_i^r$, and (3) $B_j^r \cap N(\mathbf{x}_i) \neq \varnothing$.

Lemma 5.3.27. *Suppose $\mathcal{V}$ is the Cartesian grid and c is a polynomial combination of ℓ_p functions with positive coefficients, as defined in Equation* (5.3.29)*. Then $\alpha_{\max} < \infty$.*

Proof. As shown in Theorem 5.3.24, for all appropriate $\mathbf{x}$, i, and j,

$$|\alpha_{ij}(\mathbf{x})| \leq 2 \sum_{s=1}^{\sigma} k_s \left(l d^{1/p_s}\right)^{q_s}.$$

Since this bound is independent of $\mathbf{x}$, i, and j, it follows that

$$\alpha_{\max} \leq 2 \sum_{s=1}^{\sigma} k_s \left(l d^{1/p_s}\right)^{q_s} < \infty. \qquad \square$$

Theorem 5.3.28. *If c is a norm, then $\alpha_{\max} \leq 2\delta$, where δ is the maximum ground cost for a pair of neighboring points in the grid discretizing A.*

Proof. For any iteration r, A can be covered by a finite number of boxes of width w_r. This implies that a finite number of grid points discretize A, and hence, that some finite δ exists.

Suppose $\mathbf{x} \in \bar{B}^r$. By the definition of the grid, $\mathbf{x}$ is contained in some $G = \mathrm{Conv}(S)$, where S is a finite set of neighboring grid points. For each $\mathbf{x}_a,\ \mathbf{x}_b \in S$, $\mathbf{x}_b \in N(\mathbf{x}_a)$, and hence $c(\mathbf{x}_a,\ \mathbf{x}_b) \leq \delta$. Because $\mathbf{x}_a$ and $\mathbf{x}_b$ were arbitrarily chosen, this is true of every pair of vertices of G. By the definition of G, $\mathbf{x}$ can be written as a convex combination of the points in S. Therefore, for any fixed $\mathbf{x}_0 \in S$, $c(\mathbf{x},\ \mathbf{x}_0) \leq \delta$.

Because $\mathbf{x} \in \bar{B}^r$, $\mathrm{Conv}(S) \cap B^r$ must be nonempty. Assume without loss of generality that $\mathbf{x}_0 = \mathbf{x}_i \in B_i^r$ for some $i \in \mathbb{N}_n$. By Theorem 5.3.25, there must exist a point $\mathbf{x}_j \in B_j^r$, a neighbor of $\mathbf{x}_i$, with $j \neq i$, and a point $\mathbf{b} \in A_{ij}$ such that $c(\mathbf{x}_i,\ \mathbf{b}) \leq c(\mathbf{x}_i,\ \mathbf{x}_j) \leq \delta$.

Applying the triangle inequality,

$$c(\mathbf{x},\, \mathbf{b}) \leq c(\mathbf{x},\, \mathbf{x}_i) + c(\mathbf{x}_i,\, \mathbf{b}) \leq 2\delta.$$

Therefore, $c(\mathbf{x},\, B) \leq 2\delta$. □

The squared Euclidean distance appears extensively in the literature of optimal transport. Since ℓ_2^2 does not satisfy the triangle inequality, I give a (more restrictive) result specific to this ground cost.

Theorem 5.3.29. *Let $c = \ell_2^2$, the squared Euclidean distance. If a_{ij} is estimated using $\mathbf{x} \in B_i^r$, then $|\alpha_{ij}(\mathbf{x})| \leq 2w_r\|y_j - y_i\|_1$.*

Proof. Let $\mathbf{x} = (x_1,\, \ldots,\, x_d)$, $\mathbf{y}_i = (y_1^i,\, \ldots,\, y_d^i)$, and $\mathbf{y}_j = (y_1^j,\, \ldots,\, y_d^j)$. The result in Theorem 5.3.2 proves that ℓ_2^2 satisfies the ground cost requirements and Theorem 5.3.20 implies there exists $t_* \in [0,\, 1]$ such that $\mathbf{b} = \mathbf{x} + t_* w_r \mathbf{v} \in A_{ij}$ for some $\mathbf{v} = (v_1,\, \ldots,\, v_d) \in \mathcal{V}$.

Hence, for any $k \in \mathbb{N}_d$,

$$\begin{aligned}(x_k + t_* w_r v_k - y_k^i)^2 &= (x_k - y_k^i)^2 + 2(x_k - y_k^i)t_* w_r v_k + (t_* w_r v_k)^2 \\ (x_k + t_* w_r v_k - y_k^i)^2 - (x_k - y_k^i)^2 &= 2(x_k - y_k^i)t_* w_r v_k + (t_* w_r v_k)^2 \ .\end{aligned}$$

Similarly,

$$(x_k + t_* w_r v_k - y_k^j)^2 - (x_k - y_k^j)^2 = 2(x_k - y_k^j)t_* w_r v_k + (t_* w_r v_k)^2 \ .$$

Thus, considering Equation (5.3.31) as it applies to the k-th coordinate,

$$\begin{aligned}&(x_k - y_k^i)^2 - (x_k + t_* w_r v_k - y_k^i)^2 + (x_k + t_* w_r v_k - y_k^j)^2 - (x_k - y_k^j)^2 \\ &\quad = -2(x_k - y_k^i)t_* w_r v_k + (t_* w_r v_k)^2 - \left(2(x_k - y_k^j)t_* w_r v_k + (t_* w_r v_k)^2\right)\end{aligned}$$

$$= -2(x_k - y_k^i)t_* w_r v_k + 2(x_k - y_k^j)t_* w_r v_k = 2(y_k^i - y_k^j)t_* w_r v_k.$$

Because $|t_* w_r v_k| = t_* w_r |v_k| \leq w_r$,

$$\begin{aligned}|\alpha_{ij}(\mathbf{x})| &\leq |[c(x,\, y_i) - c(b,\, y_i)] + [c(b,\, y_j) - c(x,\, y_j)]| \\ &\leq \sum_{k=1}^{d} |2t_* w_r v_k||y_k^j - y_k^i| \leq \sum_{k=1}^{d} 2w_r |y_k^j - y_k^i| = 2w_r \|y_j - y_i\|_1. \end{aligned} \qquad \square$$

Remark 5.3.30. The exact shifts $\{a_i\}_{i=1}^n$ correspond to a transport map giving the exact optimal solution of the semi-discrete problem. The approximated shifts $\{\tilde{a}_i\}_{i=1}^n$, unless equal to the exact shifts, correspond to a transport map giving the exact optimal solution to a *different* semi-discrete problem, one whose measure ν at each $\mathbf{y}_i$, $i \in \mathbb{N}_n$, corresponds to the value of $\mu(\tilde{A}_i)$.

5.3.4.4 Error bound for ground costs

In preparation for bounding the error in the Wasserstein distance, I now bound the error on the ground cost c with respect to individual points in $\bar{B}^r$.

Theorem 5.3.31. *Let $\tilde{\pi}^*$ be an approximated transport plan for the semi-discrete transport problem, with associated transport map $\widetilde{T}$. Suppose π^* is an optimal transport plan with associated map T, and let $\mathbf{x}$ in A such that $T(\mathbf{x}) = \mathbf{y}_i$, but $\widetilde{T}(\mathbf{x}) = \mathbf{y}_j$. Then the error in the ground cost at the point $\mathbf{x}$ is equal to $|g_{ij}(\mathbf{x})|$. Furthermore, if c is a polynomial combination of ℓ_p functions with positive coefficients, then there exists $\gamma_{\max}$ such that, for all $\mathbf{x} \in A$, if $T(\mathbf{x}) = \mathbf{y}_i$ and $\widetilde{T}(\mathbf{x}) = \mathbf{y}_j$ for some $i \neq j$, then*

$$|g_{ij}(\mathbf{x})| \leq \gamma_{\max} < \infty. \tag{5.3.32}$$

Proof. Let $\mathbf{x} \in A$ such that $T(\mathbf{x}) = \mathbf{y}_i$, but $\widetilde{T}(\mathbf{x}) = \mathbf{y}_j$. Then the error in the ground cost at $\mathbf{x}$ equals

$$|c(\mathbf{x},\, \mathbf{y}_i) - c(\mathbf{x},\, \mathbf{y}_j)| = |g_{ij}(\mathbf{x})|.$$

Now suppose c is a polynomial combination of ℓ_p functions with positive coefficients, defined as described in Equation (5.3.29). Because all costs are nonnegative, and achieve their maximum with $c(0,\, l\mathbb{1}_d)$, for all $\mathbf{x} \in A$,

$$|g_{ij}(\mathbf{x})| = |c(\mathbf{x},\, \mathbf{y}_i) - c(\mathbf{x},\, \mathbf{y}_j)| \leq c(0,\, l\mathbb{1}_d) = \sum_{s=1}^{\sigma} k_s \left(ld^{1/p_s}\right)^{q_s} < \infty. \qquad \square$$

If c is a norm, then $\gamma_{\max}$ can be bounded more concretely using $\alpha_{\max}$ and the set of shift differences, as shown below.

Lemma 5.3.32. *If c is a norm, then $\gamma_{\max} \leq \kappa_{\max} + 4\delta$, where*

$$\kappa_{\max} := \max_{\substack{1 \leq i < n \\ i < j \leq n}} |a_{ij}|, \tag{5.3.33}$$

and δ is the maximum ground cost for a pair of neighboring points in the grid discretizing A.

Proof. Let $\mathbf{x} \in \bar{B}^r$ and $i,\, j \in \mathbb{N}_n$ such that $i \neq j$. As a consequence of Theorems 5.3.23, 5.3.25 and 5.3.28:

$$|g_{ij}(\mathbf{x})| = |c(\mathbf{x},\, \mathbf{y}_i) - c(\mathbf{x},\, \mathbf{y}_j)| \leq |a_{ij}| + |\alpha_{ij}(\mathbf{x})| \leq \kappa_{\max} + 2c(\mathbf{x},\, \mathbf{b}) \leq \kappa_{\max} + 4\delta.$$

Because the result is independent of $\mathbf{x}$, i, and j, $\gamma_{\max} \leq \kappa_{\max} + 4\delta$. $\square$

5.3.4.5 Volume and containment for the boundary region

As shown in Section 5.3.4.4, the ground cost error for individual points is finitely bounded over a wide range of admissible ground cost functions. By definition, the measure μ is

bounded. Therefore, I propose to identify the largest possible region in which the ground cost error can be non-zero, and to show that the area of that region goes to zero as r goes to infinity. With this, I will show that the boundary method converges with respect to the Wasserstein distance.

In Equation (5.3.27), I defined a region $\bar{B}^r$ based on the point set B^r. For this, I need to know that one can choose an initial width w_1 such that, for all iterations r, $B \subset \bar{B}^r$. Fortunately, when discretizing A, one can do so with respect to the Lebesgue measure and Euclidean distance in $\mathbb{R}^d$.

Theorem 5.3.33. *There exists an initial width w_1 such that, for all w_r such that $w_r \leq w_1$, $\mathbf{x} \in \mathring{A}_i^r$ implies $\mathbf{x}^r \subseteq \mathring{A}_i$, where $\mathring{A}_i$ is the strict interior of A_i, as defined by Equation* (5.2.3).

Proof. Assume the contrary. Consider what happens as the region volumes go to zero. Given the initial assumptions, for the theorem to be false, there must be some point $\mathbf{p} \in A_i$, surrounded by an open hollow sphere in A_j of radius $\delta > 0$. Let the hollow sphere be given by

$$S := \{\mathbf{x} \in X \setminus \{\mathbf{p}\} \mid \|\mathbf{x} - \mathbf{p}\|_2 < \delta\} \subset A_j.$$

By the continuity of F, $\mathbf{p} \in A_{ij}$. Thus, $\mathbf{p}$ is a point on the level set $g_{ij}(\mathbf{x}) = a_i - a_j$. Define

$$S_g := S \setminus \{\mathbf{x} \in A \mid g_{ij}(\mathbf{x}) = a_i - a_j\}.$$

S_g is the union of two disjoint, nonempty open sets

$$S^+ := \{\mathbf{x} \in S \mid g_{ij}(\mathbf{x}) > a_i - a_j\}$$

and

$$S^- := \{\mathbf{x} \in S \mid g_{ij}(\mathbf{x}) < a_i - a_j\}.$$

However, for all $\mathbf{x}$ in the nonempty set S^-, $a_j - c(\mathbf{x}, \mathbf{y}_j) < a_i - c(\mathbf{x}, \mathbf{y}_i)$, and therefore $\mathbf{x} \notin A_j$. This contradicts that S is a subset of A_j. □

Next, I show that it is possible to choose an initial width and grid arrangement to guarantee that each point in $A^r \setminus B^r$ corresponds to a box in the interior of some region A_i.

Theorem 5.3.34. *Suppose w_1 is chosen as described in Theorem 5.3.33. Fix r, and let $w_r \leq w_1$. If $B \subseteq \bar{A}^r$, then $B \subseteq \bar{B}^r$.*

Proof. I will show the conclusion by proving that $\mathbf{x}_0 \notin \bar{B}^r$ implies $\mathbf{x}_0 \notin B$.

Suppose $\mathbf{x}_0 \notin \bar{B}^r$. If $\mathbf{x}_0 \notin \bar{A}^r$, then $\mathbf{x}_0 \notin B$, since by assumption, $B \subseteq \bar{A}^r$. Thus, assume instead that $\mathbf{x}_0 \in \bar{A}^r \setminus \bar{B}^r$.

Because $\mathbf{x}_0 \in \bar{A}^r$, $\mathbf{x}_0 \in \mathbf{x}^r$ for some $\mathbf{x} \in A^r$. There must be $\mathbf{x} \in A_i$ for some $i \in \mathbb{N}_n$, and so $\mathbf{x} \in A_i^r$. However, $\mathbf{x}_0 \notin \bar{B}^r$ implies $\mathbf{x}^r \not\subseteq \bar{B}^r$, so $\mathbf{x} \notin B^r$. Because, $\mathbf{x} \in A_i^r \setminus B^r = \mathring{A}_i^r$, by Theorem 5.3.33, $\mathbf{x}^r \subseteq \mathring{A}_i$. Hence, $\mathbf{x}_0 \in \mathring{A}_i$. Therefore, by applying Equation (5.2.3), $\mathbf{x}_0 \notin B$. □

Now that I have ensured $B \subseteq \bar{B}^r$, I aim to construct a region $\bar{B}_+^r$ with controlled volume, that encloses $\bar{B}^r$: $\bar{B}^r \subseteq \bar{B}_+^r$. Then I can show that, as $r \to \infty$, the volume of $\bar{B}_+^r$ in $\mathbb{R}^d$ goes to zero with respect to the Lebesgue measure. This allows one to put a convenient upper bound on the volume of $\bar{B}^r$ in terms of the width w_r. Because $\bar{B}_+^r$ exists solely in A, and not on the product space, one can again rely on the Euclidean distance in $\mathbb{R}^d$.

Theorem 5.3.35. *Let the region $\bar{B}_+^r \subseteq A$ be defined as*

$$\bar{B}_+^r := \{\mathbf{x} \in A \mid \|\mathbf{x} - B\|_2 \leq 2w_r\sqrt{d}\}. \tag{5.3.34}$$

For all r, $\bar{B}^r \subseteq \bar{B}_+^r$.

Proof. By definition, $\bar{B}^r \subseteq A$. Suppose $x \in \bar{B}^r$. By Theorem 5.3.28, $\|\mathbf{x} - B\|_2 \leq 2\delta$. Let $\mathbf{x}_0$ and $\mathbf{x}_1 \in N(\mathbf{x}_0)$ be elements of the r-th discretization grid of the set A. By the definition of neighbors, one can write $\mathbf{x}_1 = \mathbf{x}_0 + w_r\mathbf{v}$ for some $\mathbf{v} \in \mathcal{V}$. Thus,

$$\|\mathbf{x}_1 - \mathbf{x}_0\|_2 = w_r\|\mathbf{v}\|_2 \leq w_r\sqrt{d}.$$

Because $\mathbf{x}_0$ and $\mathbf{x}_1$ were arbitrarily chosen, this is true of every pair of neighbors. This implies $\delta \leq w_r\sqrt{d}$. Therefore, $\|\mathbf{x} - B\|_2 \leq 2w_r\sqrt{d}$, and since $\mathbf{x} \in A$, $\mathbf{x} \in \bar{B}^r_+$. $\square$

Theorem 5.3.36. *If the semi-discrete transport problem is Monge under the shift characterization, and $|B| = \tilde{L} < \infty$ with respect to the $\mathbb{R}^{d-1}$ Lebesgue measure, then there exists some $L < \infty$, such that $\left|\bar{B}^r\right| \leq w_r^d L$ with respect to the $\mathbb{R}^d$ Lebesgue measure.*

Proof. Assume that the given semi-discrete transport problem is Monge under the shift characterization, and $|B| = \tilde{L} < \infty$ with respect to the $\mathbb{R}^{d-1}$ Lebesgue measure. Then $\int_{\bar{B}^r_+} d\mathbf{x} = \int_A \chi\left[\bar{B}^r_+\right](\mathbf{x})\, d\mathbf{x}$. Let $\mathcal{B}(\mathbf{x},\, \rho)$ be the closed ball of radius ρ centered at $\mathbf{x}$, and write

$$\begin{aligned}
\int_A \chi\left[\bar{B}^r_+\right](\mathbf{x})\, d\mathbf{x} &= \int_A \chi\left[\left\{\mathbf{x} \in A \mid \|\mathbf{x} - B\|_2 \leq 2w_r\sqrt{d}\right\}\right](\mathbf{x})\, d\mathbf{x} \\
&= \int_A \chi\left[\left\{\mathbf{x} \in A \mid \mathbf{x} \in \mathcal{B}(\mathbf{z},\, 2w_r\sqrt{d}) \text{ for some } \mathbf{z} \in B\right\}\right](\mathbf{x})\, d\mathbf{x}, \\
&\leq \int_A \chi[B](\mathbf{z}) \left(\int_A \chi\left[\mathcal{B}(\mathbf{z},\, 2w_r\sqrt{d})\right](\mathbf{x})\, d\mathbf{x}\right) d\mathbf{z}.
\end{aligned}$$

For all fixed $\mathbf{x}$,

$$\int_A \chi\left[\mathcal{B}(\mathbf{x},\, 2w_r\sqrt{d})\right](\mathbf{x})\, d\mathbf{x} \leq \mathrm{Vol}_d\left(2w_r\sqrt{d}\right),$$

where $\mathrm{Vol}_d(\rho)$ is the volume of the d-dimensional sphere of radius ρ. By using the Γ function, this volume can be written as

$$\begin{aligned}
\mathrm{Vol}_d\left(2w_r\sqrt{d}\right) :=& \frac{\pi^{d/2}}{\frac{d}{2}\Gamma\left(\frac{d}{2}\right)}\left(2w_r\sqrt{d}\right)^d \\
=& \begin{cases} \frac{\pi^{d/2}}{(d/2)!}\left(2w_r\sqrt{d}\right)^d & \text{if } d = 2k \text{ for some } k \in \mathbb{Z} \\ \frac{2k!(4\pi)^k}{d!}\left(2w_r\sqrt{d}\right)^d & \text{if } d = 2k+1 \text{ for some } k \in \mathbb{Z}. \end{cases}
\end{aligned}$$

Because the volume is independent of the point $\mathbf{x} \in A$, therefore

$$\begin{aligned}\int_{\bar{B}^r_+} d\mathbf{x} &\leq \int_A \chi[B](\mathbf{z}) \int_A \chi\left[\mathcal{B}(\mathbf{z},\, 2w_r\sqrt{d})\right](\mathbf{x})\, d\mathbf{x}\, d\mathbf{z}, \\ &\leq \int_A \chi[B](\mathbf{z})\, \mathrm{Vol}_d\left(2w_r\sqrt{d}\right)\, d\mathbf{z} = \mathrm{Vol}_d\left(2w_r\sqrt{d}\right) \int_B d\mathbf{z} = w_r^d L,\end{aligned}$$

where

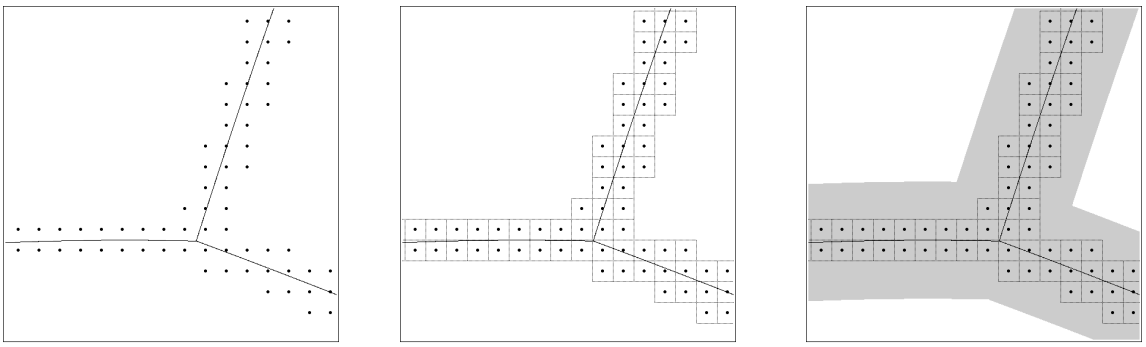
$$L := \begin{cases} \tilde{L}\frac{\pi^{d/2}}{(d/2)!}\left(2\sqrt{d}\right)^d & \text{if } d = 2k \text{ for some } k \in \mathbb{Z} \\ \tilde{L}\frac{2k!(4\pi)^k}{d!}\left(2\sqrt{d}\right)^d & \text{if } d = 2k+1 \text{ for some } k \in \mathbb{Z}. \end{cases}$$

Let $\mathbf{x} \in \bar{B}^r$. Because $\|\cdot\|_2$ is a norm, by applying Theorem 5.3.28 with $c = \ell_2$, for all $\mathbf{x} \in \bar{B}^r$, $\|\mathbf{x} - B\|_2 \leq 2\delta$. For the box grid of width w_r, $\delta = w_r\sqrt{d}$ when $c = \ell_2$. Hence, $\|\mathbf{x} - B\|_2 \leq 2w_r\sqrt{d}$, which implies $\mathbf{x} \in \bar{B}^r_+$. Thus, $\bar{B}^r \subseteq \bar{B}^r_+$, which implies $|\bar{B}^r| \leq |\bar{B}^r_+| \leq w_r^d L$. □

Remark 5.3.37. The interplay between B, B^r, $\bar{B}^r$, and $\bar{B}^r_+$ is nontrivial, and Figure 5.3 helps to visualize it properly. Figure 5.3(a) shows placement of the boundary set B^r for some $r > 1$. One can see in this image how a (very degenerate) choice of c, coupled with the right arrangement of $\mathbf{y}_i$'s, might allow a small and sharply curved boundary set to slip unnoticed between points. Section 5.4.1 discusses how to prevent this.

It is crucial that the subgrid created by B^r completely surrounds B, because that is the only way to ensure that $B \subseteq \bar{B}^r$. As Figure 5.3(b) illustrates, each point in B^r appears as the center of its corresponding box, and the boxes completely cover the boundary set.

The region $\bar{B}^r_+$ is deliberately constructed to entirely cover all the boxes in $\bar{B}^r$. As Figure 5.3(c) shows, its volume can be significantly larger than that of the boxes it contains. However, the worst-case "thickness" given to $\bar{B}^r_+$ ensures that it will always enclose both B and $\bar{B}^r$.



(a) grid B^r surrounding B (b) boxes $\bar{B}^r$ covering B (c) region $\bar{B}^r_+$ covering $\bar{B}^r$

Figure 5.3: Detail from problem in Figure 5.4(a):
Boundary set interactions near $A_0 \cap A_2 \cap A_3$

5.3.4.6 The Wasserstein distance error

In this short section I give one of the important results of this work: an error bound on the Wasserstein distance.

Theorem 5.3.38. *Suppose that $B \subseteq \bar{B}^r$, and there exists some L such that $|\bar{B}^r| = w_r^d L < \infty$ with respect to the d-dimensional Lebesgue measure. Let M be the maximum value of μ on A. If $\gamma_{\max} < \infty$ is the maximum error of the ground cost in the set $\bar{B}^r$, and $\widetilde{P}^*$ is an approximate solution created with the boundary method, then the error in the Wasserstein distance approximation $\widetilde{P}^*$ is linearly bounded with respect to w_r:*

$$\left|\widetilde{P}^* - P^*\right| \leq w_r^d L M \gamma_{\max}, \tag{5.3.35}$$

where the bound equals the maximum possible volume of $\bar{B}^r$ multiplied by the maximum value of μ and the maximum error of the ground cost.

Proof. If $\mathbf{x} \in A \setminus \bar{B}^r$, then $\mathbf{x}$ has been identified as being in the interior of A_i for some $i \in \mathbb{N}_n$. Thus, the cost error associated with the points outside $\bar{B}^r$ is zero.

Suppose instead that $\mathbf{x} \in \bar{B}^r \subseteq \bar{B}^r_+$. By definition, the absolute value of the difference between the correct and approximated ground costs at $\mathbf{x}$ is less than or equal to $\gamma_{\max}$.

Therefore, the error on the Wasserstein distance is bounded above by

$$\left|\widetilde{P}^* - P^*\right| \leq \int_{\bar{B}^r} \gamma_{\max}\, d\mu(\mathbf{x}) = \int_{\bar{B}^r} \mu(\mathbf{x})\gamma_{\max}\, d\mathbf{x}$$
$$\leq \int_{\bar{B}^r} (M)\gamma_{\max}\, d\mathbf{x} = \left|\bar{B}^r\right| M\gamma_{\max} \leq (w_r^d L) M\gamma_{\max}. \quad \square$$

Remark 5.3.39. The bounds in Theorems 5.3.35 and 5.3.38 indicate that the approximation error in computing the boundary set and the Wasserstein distance decreases linearly with the width of the boxes. This decrease is clearly also observed in practice, as Section 5.5 shows.

5.4 Computational considerations

Here I discuss some purely practical issues, such as the choice of w_1, options for exact computation of μ and the Wasserstein distance, and ways that the set of shifts can be used to reconstruct the μ-partitions.

5.4.1 Choosing w_1 to ensure $B \subseteq \bar{B}^r$

Using the information given in the initial graph, one can identify two self-evident requirements for the choice of w_1:

1. $w_1 < \min_{i,\, j \in \mathbb{N}_n} \|\mathbf{y}_i - \mathbf{y}_j\|_2$, and
2. $w_1^{-d} > n$.

Item 1 ensures that $\mathbf{y}_i,\ \mathbf{y}_j \in \mathbf{x}^r$ implies $i = j$, and Item 2 ensures that there are at least as many grid points as there are regions.

One can improve upon Item 2, ensuring that one can nearly always discard a large enough subset of A^1 to justify the expense of computation. (Otherwise, one may as well start by setting w_1 to the value intended for w_2.) In practice, this means that one wants

significantly more than n grid points, satisfying

$$w_1^{-d} \geq \omega n \qquad \text{for some } \omega > 1. \tag{5.4.1}$$

For polynomial combinations of ℓ_p functions with positive coefficients, my implementation uses $\omega = 50$, which has given consistently good results. I use a grid width derived from powers of 2, so I use $w_1 = 2^{-m}$, with m chosen to satisfy $2^{md} \geq 50n$: equivalently,

$$m = \left\lceil \frac{\ln(50n)}{d \ln 2} \right\rceil. \tag{5.4.2}$$

With this starting width, I have never observed any loss of boundary points in my tests.

5.4.2 Computing the mass $\mu(\mathbf{x}^r)$

As mentioned in Remark 5.2.3, some choices of μ may make it necessary to approximate $\mu(\mathbf{x}^r)$. In fairness, the majority of numerical studies I have seen use only simple choices of μ (most often uniform). For this reason, and to better highlight the potential of the boundary method, in this work I restricted to cases where the integral of μ over some box could be written in a closed form:

$$\mu(\mathbf{x}^r) = \int_{\mathbf{x}^r} \mu(\mathbf{z}) d\mathbf{z} = M(\mathbf{z})\Big|_{\mathbf{z}\in\mathbf{x}^r} \quad \text{with} \quad M : X \to \mathbb{R}^{\geq 0}. \tag{5.4.3}$$

Of course, since μ is a probability density function, $\int_A d\mu = 1$. For convenience, below I let $\hat{\mu}$ denote an un-normalized version of μ, and similarly for $\hat{M}$.

Remark 5.4.1. Using linearity of the integral, one can take linear combination of simple functions for which exact solutions are known. One can also construct very complex probability measures by μ-partitioning A into disjoint subsets. This case, however, requires an additional restriction in order to be sure that exact solutions can always be found: μ-partition A into subsets $S_1, \ldots, S_\sigma$, such that the boundaries of each S_s fall on the initial

set of grid lines. For each set S_s, there exists a density function $\hat{\mu}_s$ that is exactly solvable on S_s. From these, I consider $\hat{\mu}$ (and $\hat{M}$) to be the piecewise functions defined on each S_s as $\hat{\mu}_s$ (and $\hat{M}_s$, respectively).

Most of my computations were performed in two-dimensions. For such problems, given some iteration r and $\mathbf{x} = (x_1,\, x_2) \in A$, $\hat{\mu}(\mathbf{x}^r)$ can be written as

$$\begin{aligned}\hat{\mu}(\mathbf{x}^r) = = \quad & \hat{M}(x_1 + w_r/2,\, x_2 + w_r/2) - \hat{M}(x_1 + w_r/2,\, x_2 - w_r/2) \\ & - \hat{M}(x_1 - w_r/2,\, x_2 + w_r/2) + \hat{M}(x_1 - w_r/2,\, x_2 - w_r/2).\end{aligned} \tag{5.4.4}$$

The closed-form choices used in my numerical tests are shown in Table 5.1. By applying the ideas discussed in Remark 5.4.1, one can use the table entries to construct more complex measures, linear combinations with positive coefficients defined on disjoint subsets of A.

Table 5.1: Closed-form options for μ

$\hat{\mu}((x_1,\, x_2)) = 1$	$\hat{M}(u,\, v) = uv$
$\hat{\mu}((x_1,\, x_2)) = x_1^t x_2^t, \quad t > 0$	$\hat{M}(u,\, v) = \dfrac{1}{(t+1)^2}(u^{t+1}v^{t+1})$
$\hat{\mu}((x_1,\, x_2)) = e^{tx_1}, \quad t \neq 0$	$\hat{M}(u,\, v) = \dfrac{1}{t}ve^{tu}$
$\hat{\mu}((x_1,\, x_2)) = e^{tx_2}, \quad t \neq 0$	$\hat{M}(u,\, v) = \dfrac{1}{t}ue^{tv}$

5.4.3 Computing the Wasserstein distance over boxes $\mathbf{x}^r$

Remark 5.2.3 describes how the interaction between certain choices of μ and c may make it necessary to approximate $\displaystyle\int_{\mathbf{x}^r} c(\mathbf{z},\, \mathbf{y}_i)\, d\mu(\mathbf{z})$ for some $\mathbf{x}^r \subseteq A$. I performed many tests where μ could be computed exactly but the Wasserstein distance could not. In these cases, I made no attempt to approximate P^*, choosing instead to focus on the accuracy of the μ-partition generated by the approximate shift set $\{\tilde{a}_i\}_{i=1}^n$.

However, there were a number of cases in two dimensions where the choice of μ and c allowed for exact computations. In those cases, because the combination of c and μ gives

an exact solution, there exists a function $C : X \times Y \to \mathbb{R}^{\geq 0}$ such that

$$\int_{\mathbf{x}^r} c(\mathbf{z},\, \mathbf{y}_i)\, d\mu(\mathbf{z}) = C(\mathbf{z},\, \mathbf{y}_i)\Big|_{\mathbf{z}\in\mathbf{x}^r}. \tag{5.4.5}$$

As in Section 5.4.2, I write $\hat{C}$ when working with $\hat{\mu}$.

By leveraging linearity of the integral, and subdividing A into disjoint sets, similarly to Section 5.4.2, one can build several ground costs c and measures μ leading to exact computations for C (or $\hat{C}$).

For example, I performed a lot of tests in $\mathbb{R}^2$, with μ being either uniform or zero in relevant boxes. To clarify, let $X,\, Y \subset \mathbb{R}^2$, let r be some iteration, $\mathbf{x} = (x_1,\, x_2) \in A$, and $\mathbf{y} = (y_1,\, y_2) \in \{\mathbf{y}_i\}_{i=1}^n$. When $\mu(\mathbf{x}^r) = 0$, the Wasserstein distance on $\mathbf{x}^r$ is also zero. For those boxes where $\mu(\mathbf{x}^r) > 0$, one can take advantage of the uniformity to define the function $\hat{C}$ in terms of a single variable: the component-wise distance between points given by $(\Delta_1,\, \Delta_2)$, where $\Delta_1 = |x_1 - y_1|$, $\Delta_2 = |x_2 - y_2|$. When the Wasserstein distance over $\mathbf{x}^r$ can be computed and is non-zero, it takes the form

$$\begin{aligned}\int_{\mathbf{x}^r} c(\mathbf{z},\, \mathbf{y})\, d\hat{\mu}(\mathbf{z}) = \quad & \hat{C}(\Delta_1 + w_r/2,\, \Delta_2 + w_r/2) - \hat{C}(\Delta_1 + w_r/2,\, \Delta_2 - w_r/2) \\ & - \hat{C}(\Delta_1 - w_r/2,\, \Delta_2 + w_r/2) + \hat{C}(\Delta_1 - w_r/2,\, \Delta_2 - w_r/2),\end{aligned} \tag{5.4.6}$$

where $\hat{C} : \mathbb{R}^2 \to \mathbb{R}^{\geq 0}$ is some function that can be computed exactly.

Table 5.2 gives Wasserstein distance functions $\hat{C}$ for $c = \ell_2$ and all $c = \ell_p^p$ such that $p > 0$. By applying the linearity of the Wasserstein distance integral, one can compute the Wasserstein distance for linear combinations of such functions with positive coefficients.

5.4.4 Reconstructing the μ-partition from the shifts

To reconstruct the μ-partition from the shifts, one generates a full discretization of A, and uses the shifts $\{\tilde{a}_i\}_{i=1}^n$ determine the destination set A_i for each grid point.

Table 5.2: Closed-form options for C when μ is uniform or zero on A

c	$\hat{C}(u,\, v)$
ℓ_2	$\begin{cases} \frac{1}{6}u^3 \log\left(\sqrt{u^2+v^2}+v\right) & \\ \quad +\frac{1}{3}uv\sqrt{u^2+v^2} & \text{if}(u,\, v) \neq \mathbf{0} \\ \quad +\frac{1}{6}v^3 \log\left(\sqrt{u^2+v^2}+u\right) & \\ 0 & \text{if}(u,\, v) = \mathbf{0} \end{cases}$
$\ell_p^p, \quad p > 0$	$\dfrac{1}{p+1}u^{p+1}v + \dfrac{1}{p+1}uv^{p+1}$

Algorithm 5.2: Boundary reconstruction

Boundary reconstruction algorithm

(0) Set the approximated measure $\tilde{\mu}(A_i) = 0$ for all $i \in \mathbb{N}_n$, and initialize $\tilde{A}^*$.

(1) For each point $\mathbf{x} \in \tilde{A}^*$,

- Find $S_{\mathbf{x}} \subseteq \mathbb{N}_n$ such that $F(\mathbf{x}) = a_i - c(\mathbf{x},\, \mathbf{y}_i)$ for each $i \in S_{\mathbf{x}}$.
- For all $i \in S_{\mathbf{x}}$, add $\mu(\mathbf{x}^*)/s_{\mathbf{x}}$ to $\tilde{\mu}(A_i)$,

  where $s_{\mathbf{x}}$ is the number of elements in $S_{\mathbf{x}}$.

[Optional] Once the set $\{S_{\mathbf{x}} \mid \mathbf{x} \in \tilde{A}^*\}$ is computed:

(3) Construct

$$\widetilde{B}^* := \left\{ \mathbf{x} \in \tilde{A}^* \;\middle|\; \text{cardinality } |\{\, S_{\mathbf{z}} \mid \mathbf{z} \in \{\mathbf{x}\} \cup N(\mathbf{x}) \,\}| > 1 \right\}.$$

Let w_* the target width achieved by the boundary method, let $\tilde{A}^*$ be the d-dimensional grid over A with width w_*, and let $\mathbf{x}^*$ be the box of width w_* corresponding to the point $\mathbf{x} \in \tilde{A}^*$. Then the reconstruction method is summarized in Algorithm 5.2.

The approximated boundary set $\widetilde{B}^*$ generated in Step (3) is primarily useful for graphing (which is why the step is optional). If the boundary set is not needed, the neighbors need not be computed, and each $\mathbf{x} \in \tilde{A}^*$ can be discarded after computation (as can its associated set $S_{\mathbf{x}}$). Unfortunately, whether or not the set A^* is stored, the time required for boundary reconstruction is proportional to the size of the full d-dimensional grid. When w_*

is extremely small, the grid becomes so large that reconstructing the μ-partition using this method strains most computational resources.

5.5 Numerical results

In this section, I report on several numerical experiments, and discuss what they reveal about the performance and scalability of the boundary method. To offer a standard example problem that can be used to compare various choices of μ, ν, and c, I define the following semi-discrete optimal transport problem on which most of my computations are performed.

Definition 5.5.1 (The standard problem)**.** Assume the probability densities μ and ν satisfy the requirements described in Section 5.1.1. Let $X = Y = [0,\, 1]^d$, where $d = 2$ or 3, and let c be a polynomial combination of ℓ_p functions with positive coefficients, as defined in Equation (5.3.29).

For all computations, I use double-precision numbers. However, given the number of p-th root calculations in the typical problem, I assume that reliable accuracy is no better than the square root of precision. For my test machine, that equals

$$\sqrt{\texttt{eps}} := 2^{-26} \approx 1.490116 \times 10^{-8}. \tag{5.5.1}$$

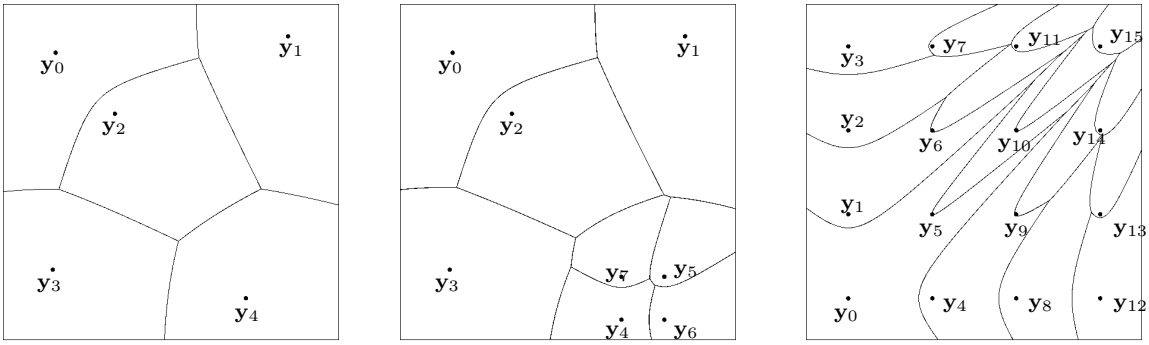
While the accuracy of some combined results, such as the Wasserstein distance, can be better than $\sqrt{\text{eps}}$, I halt most calculations when the difference between expected and actual values is less than $\sqrt{\text{eps}}$.

5.5.1 μ-partitions in $\mathbb{R}^2$

When creating μ-partitions, I prefer a discretization with target width $w_* = 2^{-11}$. Even at this level of refinement, all points and boundaries must be emphasized to aid visibility. Without it, the actual thickness of each figure's boundary line would be approximately 0.025 millimeters.

5.5.1.1 *Uniform and non-uniform measures μ and ν*

I include three examples of variations on μ and ν, shown in Figure 5.4. All three assume that the ground cost c is the Euclidean distance.



(a) μ and ν uniform (b) ν non-uniform (c) 4×4, $\mu(x_1,\, x_2) = x_1 x_2$

Figure 5.4: Partitions for problems with uniform and non-uniform measures μ and ν

Figure 5.4(a) assumes μ is the uniform continuous probability distribution on A and ν is the uniform discrete distribution with $n = 5$. The five points where $\nu = 1/5$ are placed in the positions used by Barrett and Prigozhin for their shift characterization example. As it turns out, all of the assumptions here match those of Barrett and Prigozhin. Figure 5.4(a) shows the μ-partition obtained by the boundary method; for comparison, see Barrett and Prigozhin's result in Figure 3 (right) of [5].

Starting from the points shown in Figure 5.4(a), I next take the point $\mathbf{y}_4$ and split it into four new points, each of one quarter-mass, positioned equidistantly from the point's original location. This gives a non-uniform ν with four points of weight $1/5$ and four of weight $1/20$. I kept μ uniform. The resulting μ-partition is shown in Figure 5.4(b).

In Figure 5.4(c), I chose the nonuniform probability density

$$\mu(x_1,\, x_2) = \frac{1}{4} x_1 x_2.$$

For ν, I chose the uniform 4×4 grid of points used in Figure 5.10(b). By comparing the results in Figures 5.4(c) and 5.10(b), the impact of μ's nonuniformity becomes obvious.

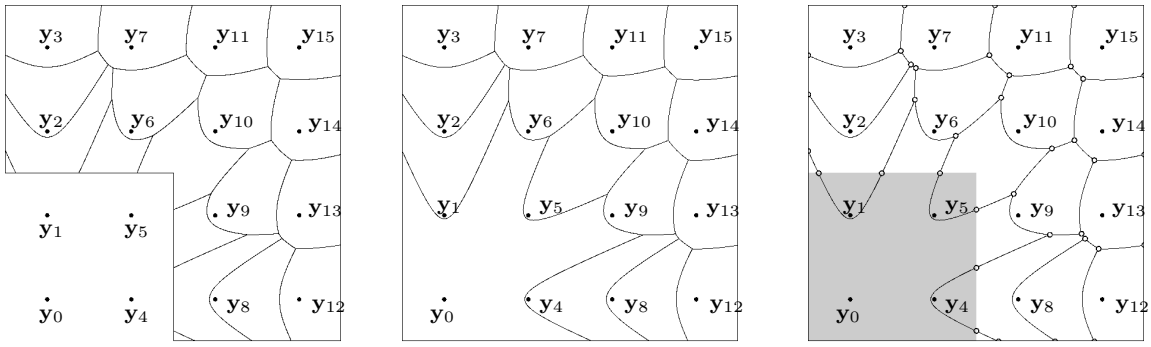
While the individual regions no longer have equal Lebesgue measure, each has equal μ-measure $1/16$. The larger regions in the lower-left correspond to the lower density of μ in that corner, while the smaller regions in the upper-right correspond to the higher concentration of μ-density there.

5.5.1.2 Discontinuous and zero-measure μ

Next, I deliberately introduce a discontinuous μ that is not strictly positive:

$$\mu(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in [0,\, 1/2]^2 \\ 4/3 & \text{otherwise.} \end{cases}$$

I still have $\int_A d\mu(\mathbf{x}) = 1$, so μ is a probability density function on $A = [0, 1]^2$. For ν, I use the same uniform 4×4 grid shown in Figure 5.10(b). Figure 5.5 shows the results.



(a) boundary region $\bar{B}^r$ (b) reconstructed problem (c) intersections and zero μ

Figure 5.5: Partition when μ is zero in the lower-left quadrant

Figure 5.5(a) shows the boundary set used to generate the solution. Points between regions are retained, as are points adjacent to the regions of measure zero. No computations are done on lower-left region, because any destination is equally valid on boxes of μ-measure zero.

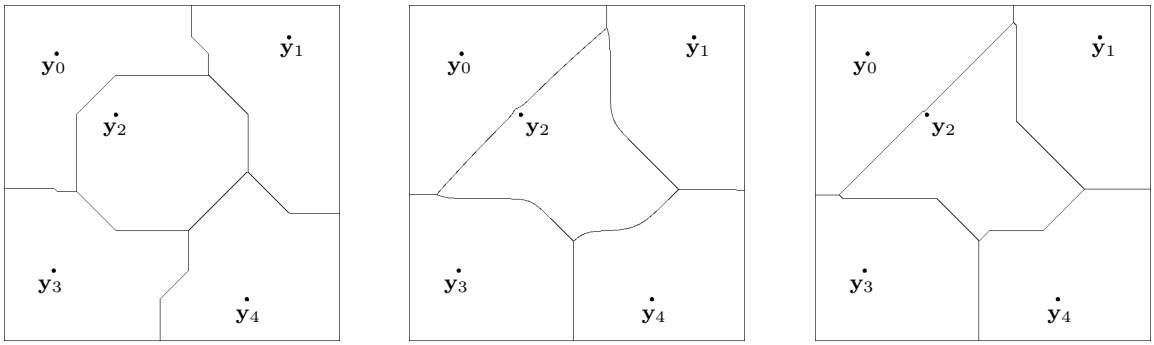
However, when the shift definition is applied to the semi-discrete optimal transport problem, there is only one valid shift-characterized solution over A. Figure 5.5(b) shows

that solution, approximated using the reconstruction process described in Section 5.4.4. Here the unique shift differences force the selection of a unique boundary set B, even in the space where μ is zero.

Figure 5.5(c) shows the reconstruction again, but here the region of μ-zero measure is shaded. This makes it easier to confirm visually that the regions do appear to have equal μ-measure, once the zero-measure quadrant is disregarded. Figure 5.5(c) also shows the locations of the intersection points identified using the boundary method. By splitting B at the given intersection points, one can partition the boundary into a set of simple smooth curves in $\mathbb{R}^2$.

5.5.1.3 Norms as ground cost functions

The above computations all assume the ground cost function equals the Euclidean distance. In fact, one can apply the boundary method to a wide range of polynomial ℓ_p ground costs. Using Barrett and Prigozhin's problem, shown for the Euclidean distance in Figure 5.4(a), μ-partitions can be generated for a wide range of polynomial ℓ_p ground costs. Three results, for the ℓ_1, ℓ_{10}, and ℓ_∞ norms, are shown in Figure 5.6.

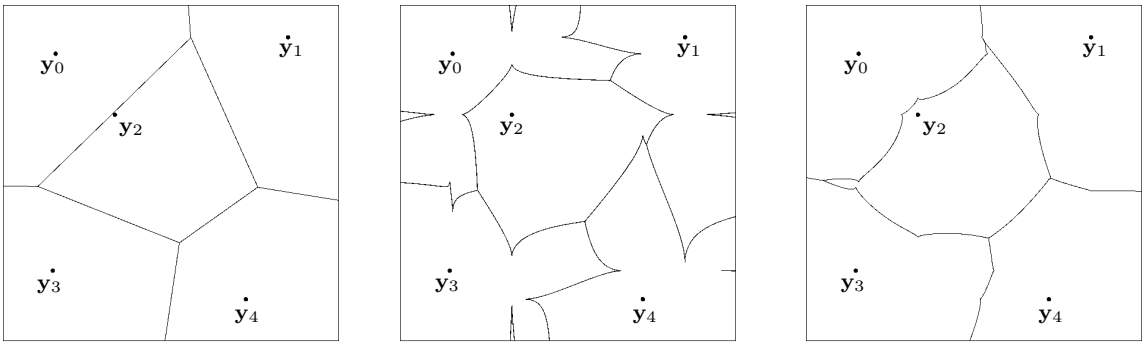


(a) ℓ_1 ground cost (b) ℓ_{10} ground cost (c) $c = \ell_\infty$ norm

Figure 5.6: μ-partitions of equal area
using different ground cost norms

5.5.1.4 *Other ground costs c*

The examples above all assume the ground cost function is a norm, and hence is well-behaved, in the sense of satisfying basic properties such as the triangle inequality and absolute homogeneity. However, my results indicate that the boundary method works equally well on much more general ground cost functions. Three examples are shown in Figure 5.7.



(a) ℓ_2^2 ground cost (b) $\ell_{1/2}$ ground cost (c) $c = 4\ell_2^{28/5} + 61\ell_{1/2}$

Figure 5.7: Partitions of equal area
with non-norm ground cost functions

Figure 5.7(a), on the left, is the result given by ℓ_2^2. The squared-Euclidean distance is not a norm, because it fails to satisfy the triangle inequality. As a result, I was only able to make the most general mathematical claims regarding its behavior; e.g., see Theorem 5.3.29. However, the boundary method is able to compute it effortlessly. Furthermore, as Tables 5.9 and 5.10 show, its convergence behavior is practically identical to that of the norms ℓ_1 and ℓ_2.

As Figure 5.7(b) illustrates, even ℓ_p costs with $p < 1$ can be approximated; when $p < 1$, the regions become discontinuous, as typified by the "spikes" on the exterior walls. (The spike on the lower right is part of the region coupled with $\mathbf{y}_3$, while the other four spikes are coupled with $\mathbf{y}_2$.) Of course, $\ell_{1/2}$ is *not* a norm; in fact, $\ell_{1/2}$ is a concave function. This is a particularly valuable result, given that concave functions are used to solve transport problems involving economies of scale; see [63].

Figure 5.7(c) shows a ground cost function defined as a polynomial combination of ℓ_p functions with positive coefficients:

$$c = 4\ell_2^{28/5} + 61\ell_{1/2}.$$

This ground cost function is neither convex nor concave, changing behavior depending on distance.

5.5.2 μ-partitions in $\mathbb{R}^3$

As I showed in Section 5.3, there is no theoretical obstacle to applying the boundary method to higher-dimensional problems. The main difficulty that arises is the practical issue of presenting a complex, high-dimensional space in lower dimensions. Take the example shown in Figures 5.8 and 5.9.

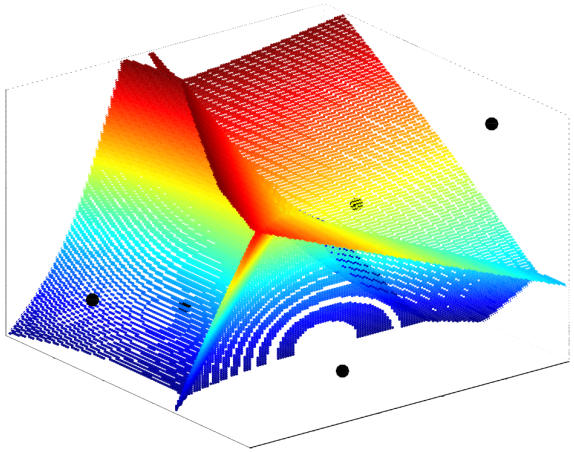


Figure 5.8: Three-dimensional semi-discrete solution with $n = 5$

The image in Figure 5.8 was generated by taking $c = \ell_2$, μ the uniform continuous probability density, and ν the uniform discrete probability density with five randomly-placed non-zero points in $[0,\, 1]^3$. Even in this relatively simple case, it was impossible to find a single point-of-view that clearly showed all five non-zero points, while clearly illustrating the boundaries of the μ-partitions. The image chosen shows three points clearly and two

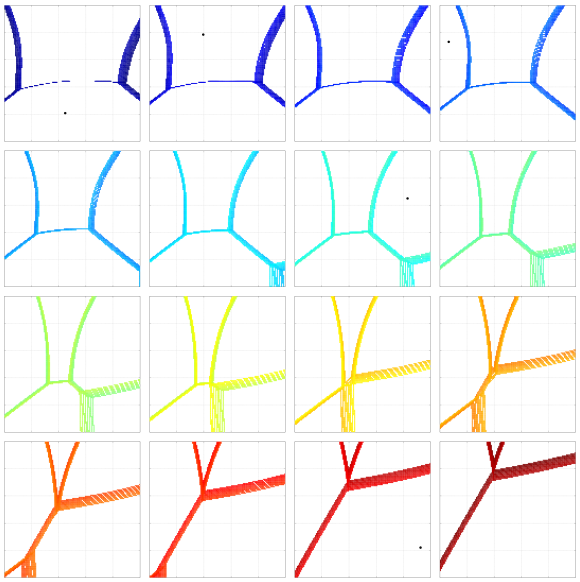


Figure 5.9: Cross-sections of Figure 5.8

more partially obscured (on the left and right center). Even though I made an effort to keep the μ-partition boundaries semi-transparent, only three of the regions are clearly visible. The shapes of the other two are at least partially obscured by their own boundary surfaces.

However, even though clear illustration remains a problem, from the point of view of the boundary method the computations were successful. I was able to identify the boundary set and the set of shifts, and use them to successfully reconstruct the μ-partition.

5.5.3 Accuracy of the Wasserstein distance

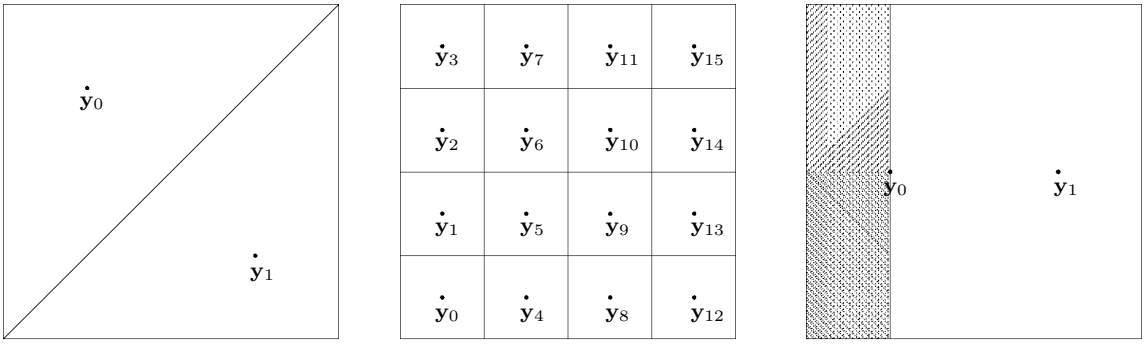
One way to consider the accuracy and convergence properties of the Wasserstein distance is to determine such distances with different choices of w_*, and compare those results, both to each other and to the exact distance (if known). To distinguish the approximated Wasserstein distance with different choices of w_*, I use the notation

$$\widetilde{P}^*_m, \qquad \text{where } m = \log_2 w_*. \tag{5.5.2}$$

Since my computations use widths of 2^{-m} for integer values of m, this gives an easy way to distinguish such approximations.

5.5.3.1 *When exact values are known.*

If one knows the exact Wasserstein distance for the semi-discrete problem, then one can compute the difference and determine an actual error value. When μ is uniform, I use the formula for $c = \ell_2$, given in Table 5.2, to compute exact values for the two problems shown in Figures 5.10(a) and 5.10(b), and the formula for $c = \ell_1$ in Table 5.2 to compute the value for the problem given in Figure 5.10(c).



(a) 2 collinear points (b) 4×4 grid arrangement (c) non-Monge

Figure 5.10: Problems where the exact Wasserstein distance and set of shifts are known

For two points on the Northwest-Southeast diagonal, placed as shown in Figure 5.10(a), and ν is uniform, the exact Wasserstein distance is equal to

$$\begin{aligned} P^*_{\text{NWSE}} :=& \frac{1}{96}\left[\sqrt{2} + 7\sqrt{10} + \sinh^{-1}(1) + 2\sqrt{2}\sinh^{-1}(2) + \sinh^{-1}(3)\right] \\ \approx& 0.3159707808963017. \end{aligned} \tag{5.5.3}$$

Table 5.3 shows the approximated Wasserstein distance, $\widetilde{P}^*$, for various choices of w_*. It also shows the error, both in absolute terms and as a percentage of the actual Wasserstein distance. As my results show, the actual decrease in error is roughly quadratic in w_*:

$$|\widetilde{P}^* - P^*| \approx 2.122(w_*)^{1.995}. \tag{5.5.4}$$

Table 5.3: Wasserstein approximation and error values for the NWSE problem

w_*	$\widetilde{P}^*$	abs. error	% error
2^{-5}	0.318069754980	2.10×10^{-3}	6.64×10^{-1} %
2^{-6}	0.316503329020	5.33×10^{-4}	1.69×10^{-1} %
2^{-7}	0.316104858309	1.34×10^{-4}	4.24×10^{-2} %
2^{-8}	0.316004415577	3.36×10^{-5}	1.06×10^{-2} %
2^{-9}	0.315979203843	8.42×10^{-6}	2.67×10^{-3} %
2^{-10}	0.315972888409	2.11×10^{-6}	6.67×10^{-4} %
2^{-11}	0.315971307996	5.27×10^{-7}	1.67×10^{-4} %
2^{-12}	0.315970912699	1.32×10^{-7}	4.17×10^{-5} %
2^{-13}	0.315970813850	3.30×10^{-8}	1.04×10^{-5} %
2^{-14}	0.315970789135	8.24×10^{-9}	2.61×10^{-6} %
2^{-15}	0.315970782956	2.06×10^{-9}	6.52×10^{-7} %
2^{-16}	0.315970781411	5.15×10^{-10}	1.63×10^{-7} %
2^{-17}	0.315970781025	1.29×10^{-10}	4.07×10^{-8} %

When I have a 4×4 arrangement of boxes, with each $\mathbf{y}_i$ in the center, as shown in Figure 5.10(b), when ν is uniform the exact Wasserstein distance is equal to

$$P^*_{4\times 4} := \frac{1}{24}\left[\sqrt{2} + \sinh^{-1}(1)\right] \approx 0.09564946455802659. \tag{5.5.5}$$

Table 5.4 shows the approximated Wasserstein distance, $\widetilde{P}^*$, for various choices of w_*. It also shows the error, both in absolute terms and as a percentage of the actual Wasserstein distance. Again, the observed error is roughly quadratic in w_*:

$$|\widetilde{P}^* - P^*| \approx 5.254(w_*)^{1.999}. \tag{5.5.6}$$

For the arrangement of points given in Figure 5.10(c), when $c = \ell_1$ and $\nu(\mathbf{y}_0) \leq 1/4$, $A_0 \subset A_1$. The shading effect visible on the left hand side of Figure 5.10(c) is a consequence of the sets' overlap on reconstruction process.

Because $A_0 \subset A_1$, the semi-discrete transport problem is not Monge under the shift characterization. As a consequence, the transport uniqueness conditions of Theorem 5.3.10 and the Wasserstein distance convergence conditions of Theorem 5.3.38 do not apply.

Table 5.4: Wasserstein approximation and error values for the 4×4 problem

w_*	$\widetilde{P}^*$	abs. error	% error
2^{-5}	0.100754632697	5.11×10^{-3}	5.34×10^{-0} %
2^{-6}	0.096936833591	1.29×10^{-3}	1.35×10^{-0} %
2^{-7}	0.095972001451	3.23×10^{-4}	3.37×10^{-1} %
2^{-8}	0.095730142233	8.07×10^{-5}	8.43×10^{-2} %
2^{-9}	0.095669636693	2.02×10^{-5}	2.11×10^{-2} %
2^{-10}	0.095654507762	5.04×10^{-6}	5.27×10^{-3} %
2^{-11}	0.095650725370	1.26×10^{-6}	1.32×10^{-3} %
2^{-12}	0.095649779762	3.15×10^{-7}	3.30×10^{-4} %
2^{-13}	0.095649543359	7.88×10^{-8}	8.24×10^{-5} %
2^{-14}	0.095649484258	1.97×10^{-8}	2.06×10^{-5} %
2^{-15}	0.095649469483	4.93×10^{-9}	5.15×10^{-6} %
2^{-16}	0.095649465789	1.23×10^{-9}	1.29×10^{-6} %
2^{-17}	0.095649464866	3.08×10^{-10}	3.22×10^{-7} %

If $\nu(\mathbf{y}_0) = 1/8$ and $\nu(\mathbf{y}_1) = 7/8$, the exact Wasserstein distance is equal to

$$P^*_{\text{non-Monge}} := \frac{1}{2}. \tag{5.5.7}$$

Table 5.5 shows the approximated Wasserstein distance, $\widetilde{P}^*$, for various choices of w_*. It also shows the error, both in absolute terms and as a percentage of the actual Wasserstein distance. The decrease in error is still roughly quadratic in w_*:

$$|\widetilde{P}^* - P^*| \approx 0.238(w_*)^{1.995}. \tag{5.5.8}$$

For all three problems, I also know the exact shift values. In the 4×4 problem, since every point in A goes to the nearest $\mathbf{y}_i$, the shift differences are all zero, which means every shift must be identical. It turns out that the shift values are identical for every choice of w_*. This is a result of computing regions that exactly correspond to the structure of the grid.

The shift values for the NWSE problem are more interesting. Once again, every point in A goes to the nearest $\mathbf{y}_i$, so the shift differences are all zero, which means every shift must be identical. However, this time the values obtained are inexact, and the error changes with the choice of w_*. Because the NWSE problem has only two shifts, and one is fixed,

Table 5.5: Wasserstein approximation and error values for the non-Monge problem

w_*	$\widetilde{P}^*$	abs. error	% error
2^{-5}	0.4997711181641	2.29×10^{-4}	4.58×10^{-2} %
2^{-6}	0.4999408721924	5.91×10^{-5}	1.18×10^{-2} %
2^{-7}	0.4999849796295	1.50×10^{-5}	3.00×10^{-3} %
2^{-8}	0.4999962151051	3.78×10^{-6}	7.57×10^{-4} %
2^{-9}	0.4999990500510	9.50×10^{-7}	1.90×10^{-4} %
2^{-10}	0.4999997620471	2.38×10^{-7}	4.76×10^{-5} %
2^{-11}	0.4999999404536	5.95×10^{-8}	1.19×10^{-5} %
2^{-12}	0.4999999851061	1.49×10^{-8}	2.98×10^{-6} %
2^{-13}	0.4999999962756	3.72×10^{-9}	7.45×10^{-7} %
2^{-14}	0.4999999990688	9.31×10^{-10}	1.86×10^{-7} %
2^{-15}	0.4999999997672	2.33×10^{-10}	4.66×10^{-8} %
2^{-16}	0.4999999999418	5.82×10^{-11}	1.16×10^{-8} %
2^{-17}	0.4999999999854	1.46×10^{-11}	2.91×10^{-9} %

I have only a single set of error values, corresponding to the difference between the two. Percentages are meaningless, since the actual value is arbitrarily chosen, so I present only the absolute error. As Table 5.6 shows, the decrease in shift error is roughly linear with respect to w_*:

$$|\tilde{a}_2 - \tilde{a}_1| \approx 0.339(w_*)^{1.008}. \tag{5.5.9}$$

Table 5.6: Shift error values for the NWSE problem

w_*	abs. shift error
2^{-4}	2.18×10^{-2}
2^{-5}	1.04×10^{-2}
2^{-6}	5.07×10^{-3}
2^{-7}	2.50×10^{-3}
2^{-8}	1.24×10^{-3}
2^{-9}	6.20×10^{-4}
2^{-10}	3.09×10^{-4}
2^{-11}	1.55×10^{-4}
2^{-12}	7.72×10^{-5}
2^{-13}	3.86×10^{-5}
2^{-14}	1.93×10^{-5}
2^{-15}	9.65×10^{-6}
2^{-16}	4.83×10^{-6}

In the non-Monge problem, the shift values must satisfy the relation $a_1 = a_0 + 1/2$. This is a direct consequence of $A_0 \subset A_1$ and the fact that $c(\mathbf{y}_0, \mathbf{y}_1) = 1/2$. In computation, the shift values exactly satisfy this relation for every choice of w_*.

5.5.3.2 When exact values are not known.

If the exact solution is unknown, one can still get a sense of the convergence properties by looking at changes in the error bounds and convergence behavior of the Wasserstein distance approximation as $w_* \to 0$.

As Section 5.3.4.6 shows, even if the Wasserstein distance is unknown, the Wasserstein approximation error at the end of the r-th iteration is bounded above by

$$\sum_{\mathbf{x} \in B^r} \mu(\mathbf{x}^r) \max_{\mathbf{x}_0 \in \mathbf{x}^r} g_{ij}(\mathbf{x}_0), \tag{5.5.10}$$

where i and j, $i \neq j$ refer to the destinations of $\mathbf{x}$ and some neighbor.

As $w_r \to 0$, $\max_{\mathbf{x}_0 \in \mathbf{x}^r} g_{ij}(\mathbf{x}_0) \to |a_{ij}|$ for each $i \neq j$, and $\mu(\bar{B}^r) \to 0$. Hence, if the boundary method is working effectively, at worst one can expect to see these error values decreasing linearly with respect to $\mu(\bar{B}^r)$. If μ is a norm, the property of absolute homogeneity implies that this relationship is equivalent to linear convergence in w_*.

I considered the change in the approximated Wasserstein distance for the problem of Barrett and Prigozhin using three canonical ℓ_p ground costs: ℓ_1, ℓ_2, and ℓ_2^2. The resulting μ-partitions are shown in Figures 5.6(a), 5.4(a), and 5.7(a), respectively.

For each ground cost, I considered the Wasserstein approximation error in two ways:

1. The worst-case error bound given by applying Equation (5.5.10).
2. The rate of change with respect to a fixed approximation,

$$\Delta \widetilde{P}^*_{m_*}(w_*) := \Delta \widetilde{P}^*_{m_*}(2^{-m}) = \frac{\left| \widetilde{P}^*_m - \widetilde{P}^*_{m_*} \right|}{\left| \widetilde{P}^*_{m+1} - \widetilde{P}^*_{m_*} \right|}, \qquad \text{where } m > m_* - 1. \tag{5.5.11}$$

I present results using $m_* = 16$, my minimum computed Wasserstein approximation, but different widths showed similar behavior.

Each table of data shows the widths, approximated Wasserstein distances, worst-case error bounds, difference from the approximation at $w_* = 2^{-16}$, and rate of change. Table 5.7 presents the ℓ_1 data, Table 5.8 the data for ℓ_2, and Table 5.9 the data for ℓ_2^2.

Table 5.7: Wasserstein distance approximation with ℓ_1 ground cost

w_*	$\widetilde{P}^*$	worst-case error	$\lvert\widetilde{P}^*_m - \widetilde{P}^*_{16}\rvert$	$\Delta\widetilde{P}^*_{16}$
2^{-4}	0.27806417174	3.04×10^{-2}	2.10×10^{-2}	3.85
2^{-5}	0.26248877574	1.34×10^{-2}	5.47×10^{-3}	4.22
2^{-6}	0.25831759825	6.17×10^{-3}	1.29×10^{-3}	3.97
2^{-7}	0.25734903449	3.12×10^{-3}	3.26×10^{-4}	3.98
2^{-8}	0.25710459604	1.52×10^{-3}	8.20×10^{-5}	5.89
2^{-9}	0.25703652819	7.62×10^{-4}	1.39×10^{-5}	2.27
2^{-10}	0.25702874864	3.81×10^{-4}	6.13×10^{-6}	3.74
2^{-11}	0.25702425853	1.96×10^{-4}	1.64×10^{-6}	5.17
2^{-12}	0.25702293819	9.23×10^{-5}	3.16×10^{-7}	1.18
2^{-13}	0.25702289102	4.73×10^{-5}	2.69×10^{-7}	6.60
2^{-14}	0.25702258102	2.33×10^{-5}	4.08×10^{-8}	6.44
2^{-15}	0.25702262814	1.14×10^{-5}	6.33×10^{-9}	—
2^{-16}	0.25702262181	5.68×10^{-6}	$0.00 \times 10^{+0}$	—

Table 5.8: Wasserstein distance approximation with ℓ_2 ground cost

w_*	$\widetilde{P}^*$	worst-case error	$\lvert\widetilde{P}^*_m - \widetilde{P}^*_{16}\rvert$	$\Delta\widetilde{P}^*_{16}$
2^{-4}	0.22192212892	2.39×10^{-2}	1.44×10^{-2}	2.17
2^{-5}	0.21116941922	1.10×10^{-2}	3.62×10^{-3}	4.06
2^{-6}	0.20843843218	5.30×10^{-3}	8.92×10^{-4}	3.73
2^{-7}	0.20778555920	2.66×10^{-3}	2.39×10^{-4}	4.39
2^{-8}	0.20760061927	1.32×10^{-3}	5.46×10^{-5}	3.90
2^{-9}	0.20756003780	6.60×10^{-4}	1.40×10^{-5}	4.06
2^{-10}	0.20754950067	3.30×10^{-4}	3.44×10^{-6}	3.94
2^{-11}	0.20754693188	1.65×10^{-4}	8.72×10^{-7}	3.88
2^{-12}	0.20754628427	8.25×10^{-5}	2.25×10^{-7}	4.18
2^{-13}	0.20754611339	4.13×10^{-5}	5.38×10^{-8}	4.17
2^{-14}	0.20754607249	2.06×10^{-5}	1.29×10^{-8}	5.28
2^{-15}	0.20754606205	1.03×10^{-5}	2.44×10^{-9}	—
2^{-16}	0.20754605961	5.16×10^{-6}	$0.00 \times 10^{+0}$	—

Table 5.9: Wasserstein distance approximation with ℓ_2^2 ground cost

w_*	$\widetilde{P}^*$	worst-case error	$\lvert\widetilde{P}^*_m - \widetilde{P}^*_{16}\rvert$	$\Delta\widetilde{P}^*_{16}$
2^{-4}	0.06181266150	1.46×10^{-2}	8.91×10^{-3}	3.91
2^{-5}	0.05518701361	6.75×10^{-3}	2.28×10^{-3}	3.94
2^{-6}	0.05348575076	3.28×10^{-3}	5.79×10^{-4}	4.22
2^{-7}	0.05304403010	1.61×10^{-3}	1.37×10^{-4}	3.86
2^{-8}	0.05294236638	8.08×10^{-4}	3.55×10^{-5}	3.92
2^{-9}	0.05291589325	4.03×10^{-4}	9.07×10^{-6}	4.27
2^{-10}	0.05290894978	2.02×10^{-4}	2.12×10^{-6}	3.80
2^{-11}	0.05290738369	1.01×10^{-4}	5.59×10^{-7}	4.14
2^{-12}	0.05290695997	5.04×10^{-5}	1.35×10^{-7}	4.24
2^{-13}	0.05290685675	2.52×10^{-5}	3.19×10^{-8}	3.98
2^{-14}	0.05290683286	1.26×10^{-5}	8.01×10^{-9}	5.56
2^{-15}	0.05290682630	6.30×10^{-6}	1.44×10^{-9}	—
2^{-16}	0.05290682486	3.15×10^{-6}	$0.00 \times 10^{+0}$	—

In all three cases, the worst-case error is roughly linear in w_*, and the rate of change $\Delta\widetilde{P}^*_{16}$ is roughly quadratic in w_*. Specific equations are given in Table 5.10

Table 5.10: Wasserstein approximation behavior with respect to w_*

$c = \ell_1$	$\mathrm{err}_{\max}(w_*) \approx 0.457(w_*)^{1.020}$	$\Delta\widetilde{P}^*_{16}(w_*) \approx 1.186(w_*)^{2.025}$
$c = \ell_2$	$\mathrm{err}_{\max}(w_*) \approx 0.361(w_*)^{1.008}$	$\Delta\widetilde{P}^*_{16}(w_*) \approx 4.151(w_*)^{2.023}$
$c = \ell_2^2$	$\mathrm{err}_{\max}(w_*) \approx 0.221(w_*)^{1.008}$	$\Delta\widetilde{P}^*_{16}(w_*) \approx 2.668(w_*)^{2.029}$

5.5.4 Reconstruction and shift accuracy

As described in Section 5.4.4, when the finest discretization used by the boundary method is small enough to be created over all A, one can use the approximate shifts $\{\tilde{a}_i\}_{i=1}^n$ to reconstruct the full μ-partition on a grid of that size. When this is possible, one can evaluate the accuracy of the shifts in terms of their ability to successfully reconstruct the correct μ-partition.

One can visually compare the boundary set approximated by the reconstruction to that output by the boundary method. While this is not a rigorous evaluation method, looking at how well the two match helps one see how accurately the shifts were computed, and

where in A the inaccuracies are felt. This review helps reveal the geometry underlying the semi-discrete transport problem.

More rigorously, one can evaluate the accuracy of the shifts in terms of their ability to recreate the measure ν. For each $i \in \mathbb{N}_n$, $\nu(\mathbf{y}_i)$ is known and the optimal transport plan must give $\mu(A_i) = \nu(\mathbf{y}_i)$. Hence, one can compute the value of $\mu(\tilde{A}_i)$ in the reconstruction for each $i \in \mathbb{N}_n$ and compute the maximum error in the μ-partition volumes, given by

$$\nu_{\text{err}} := \max_{i \in \mathbb{N}_n} |\mu(\tilde{A}_i) - \nu(\mathbf{y}_i)|. \tag{5.5.12}$$

The reconstruction is discrete, so in fact it only approximates the actual $\mu(\tilde{A}_i)$. However, one can compute this error, as described in Section 5.4.4, so one could disregard any error small enough to be blamed on the discretization. In practice, the ν-reconstruction error never seems to be small enough for the discretization error to be relevant.

As a consequence of the results in Section 5.3.4.3, one can expect shift approximations to converge linearly when c satisfies the triangle inequality. In practice, I see linear convergence of the shift values, regardless of the choice of c. However, the convergence behavior of the maximum error in μ-partition values, ν_{err}, is highly nonlinear, and appears to depend on the structure of the ground cost function and the measures μ and ν. Because of this nonlinearity, the extra effort to compute the shifts using multiple intersection points reaps significant dividends here. (See Section 5.2.2.2 for details on how to do this.)

Table 5.11 gives ν_{err} for widths ranging from 2^{-4} to 2^{-13}, with different cost functions. All results are based on computation of the Barrett and Prigozhin example, so $\nu(\mathbf{y}_i) = 1/5$ for each $\mathbf{y}_i$. Visual depictions of the solution μ-partitions for the cost functions $c = \ell_2^2$, $c = \ell_2$, $c = \ell_1$, and $c = \ell_{1/2}$ are shown in Figures 5.7(a), 5.4(a), 5.6(a), and 5.7(b), respectively.

For all four costs depicted in Table 5.11, the scaling of ν_{err} is roughly linear with respect to w_*, but there is a great deal of variation in the data. When the scaling is assumed to be

Table 5.11: ν_{err} with respect to w_*
for different cost functions

w_*	$c = \ell_2^2$	$c = \ell_2$	$c = \ell_1$	$c = \ell_{1/2}$
2^{-4}	2.81×10^{-2}	1.74×10^{-2}	1.05×10^{-2}	2.36×10^{-2}
2^{-5}	1.96×10^{-2}	4.59×10^{-3}	4.35×10^{-3}	2.05×10^{-2}
2^{-6}	7.07×10^{-3}	2.80×10^{-3}	4.93×10^{-3}	2.31×10^{-3}
2^{-7}	2.89×10^{-3}	6.90×10^{-4}	1.10×10^{-3}	3.71×10^{-3}
2^{-8}	1.94×10^{-3}	1.30×10^{-3}	1.35×10^{-3}	2.06×10^{-3}
2^{-9}	8.95×10^{-4}	5.20×10^{-4}	7.14×10^{-4}	2.68×10^{-4}
2^{-10}	2.58×10^{-4}	2.43×10^{-4}	4.23×10^{-4}	6.78×10^{-4}
2^{-11}	1.51×10^{-4}	1.22×10^{-4}	1.04×10^{-4}	5.76×10^{-4}
2^{-12}	1.74×10^{-4}	4.32×10^{-5}	1.09×10^{-4}	5.25×10^{-4}
2^{-13}	6.52×10^{-5}	2.89×10^{-5}	1.12×10^{-5}	6.33×10^{-4}

linear, I get the equations and R^2 values shown in Table 5.12. The relatively low R^2 values (0.920–0.979) are caused by the nonlinear behavior of the shift convergence. For example, take $c = \ell_1$, which is 1.04×10^{-4} when $w_* = 2^{-11}$, increases slightly to 1.09×10^{-4} when $w_* = 2^{-12}$, and then decreases by nearly a factor of 10, to 1.12×10^{-5}, when $w_* = 2^{-13}$.

Table 5.12: ν_{err} equations with respect to w_*

$c = \ell_2^2$	$\nu_{\text{err}}(w_*) \approx 0.482(w_*)$	$R^2 = 0.979$
$c = \ell_2$	$\nu_{\text{err}}(w_*) \approx 0.247(w_*)$	$R^2 = 0.951$
$c = \ell_1$	$\nu_{\text{err}}(w_*) \approx 0.170(w_*)$	$R^2 = 0.957$
$c = \ell_{1/2}$	$\nu_{\text{err}}(w_*) \approx 0.421(w_*)$	$R^2 = 0.920$

5.5.5 Scaling behavior

One important advantage to the boundary method is its reduction of the complexity of the discretized problem, compared to traditional methods. Before considering the numerical

results, it is worth developing a generalized comparison that puts this reduction in perspective:

Suppose for the sake of argument that a discretization with width 2^{-M} is required to solve a problem with N positive points in Y. Generating the full grid would create a product space $X \times Y$ of size $2^{2M} \cdot N$. Say the boundary method is used instead, with a fixed initial discretization width of 2^{-4}. Each application of Step (2) of the boundary method algorithm removes approximately half the points in A^r, so by discarding interiors the method constructs a product space of size $2^{M+4} \cdot N$.

To consider how this affects the cost, assume for the moment that one computes solutions for both the boundary method and the full product space using a typical linear solver: the network simplex method. As proved by Tarjan in [106], the worst-case complexity of the network simplex is $\mathcal{O}(VE \log V \log CV)$, where V is the number of vertices, E the number of edges, and C the maximum possible ground cost value. If C is fixed, as it is for purposes of this comparison, the complexity is $\mathcal{O}(VE(\log V)^2)$.

Using the network simplex method to solve the largest boundary problem of size $2^{M+4} \cdot 5$ gives $V = 2^{M+4} + N$ and $E = 2^{M+4} \cdot N$, so the complexity gives

$$VE(\log V)^2 \approx 2^{M+4} \cdot (2^{M+4} \cdot N) \cdot (M+4)^2 \approx 2^{2M+8} \cdot (M+4)^2 N.$$

Because of the reiterative nature of the boundary method, multiply this by its logarithm,

$$\log(2^{2M+8} \cdot (M+4)^2 N) \approx 2(M + 4 + \log M),$$

giving

$$2^{2M+9} \cdot (M+4)^2 N(M + \log M + 4).$$

By comparison, solving over the full product space gives $V = 2^{2M} + N$ and $E = 2^{2M} \cdot N$, so

$$VE(\log V)^2 \approx 2^{2M} \cdot (2^{2M} \cdot N) \cdot (2M)^2 \approx 2^{4M+2} \cdot M^2 N.$$

Hence, all other things being equal, the ratio of the two approaches is

$$2^{4M+2-(2M+9)} \cdot \frac{M^2}{(M+4)^2(M + \log M + 4)}$$

approximately 2^{2M} to M. Even if one assumes a solver with complexity $\mathcal{O}(V)$ (and no such solver exists), the ratio would be approximately 2^M to M.

Of course, the improved complexity would be irrelevant if the constant factor was excessively large. Fortunately, that this is not the case, as the next section illustrates.

5.5.5.1 Scaling on the plane with respect to $W = 1/w_*$

One can expect the time required to solve will be proportional to the number of boundary points, and the number of boundary points should be related to the size of the discretization, which is inversely proportional to some power of w_*. To facilitate understanding this relationship, first consider scaling on the plane with respect to

$$W = \frac{1}{w_*}. \tag{5.5.13}$$

The process of evaluating scaling with respect to W is fairly straight-forward: choose a target problem, and consider it with various values of w_*.

For this target, take the Barrett and Prigozhin example in $[0,\, 1]^2$. Let μ and ν be uniform, take $c = \ell_2$, and fix the locations of the 5 points where $\nu = 1/5$, as depicted in Figure 5.4(a). Define a target width $w_* = 2^{-m}$ for some $m \in \mathbb{N}$, and compute the time taken by the boundary method. Then increment m and do it again. By repeating this pro-

cess for a few different location sets (and averaging them), one can estimate the average scaling behavior of the boundary method with respect to W. The results of this scaling test are shown in Table 5.13. Scaling in the plane with respect to W, as shown in Table 5.13, is approximated by the equations shown on the left side of Table 5.14.

Table 5.13: Scaling with respect to W

W	Time (sec)	Store (MB)
2^4	0.001	0.081
2^5	0.003	0.176
2^6	0.007	0.369
2^7	0.016	0.751
2^8	0.035	1.524
2^9	0.075	3.056
2^{10}	0.162	6.127
2^{11}	0.375	12.260
2^{12}	0.855	24.540
2^{13}	2.005	49.100
2^{14}	4.497	98.210
2^{15}	11.025	196.400
2^{16}	28.093	394.400
2^{17}	60.577	785.800
2^{18}	132.397	1571.840
2^{19}	292.158	3151.872
2^{20}	640.660	6309.888

Table 5.14: Time and memory scaling with respect to W alone

Time	$T(W) \approx 4.356 \times 10^{-5} W \ln W$	$R^2 = 0.999$
Storage	$S(W) \approx 6.015 \times 10^{-3} W$	$R^2 = 1.000$

5.5.5.2 Scaling on the plane with respect to $N = \max n$

To evaluate planar scaling with respect to N, I performed multiple runs in $[0,\, 1]^2$ where $W = 2^{11}$ was fixed and μ and ν were uniform, but the $N = n$ points where $\nu = 1/n$ were placed at random locations in A. Because the resulting time data was highly dependent on point placement, it was extremely noisy. Thus, I did ten runs for each N and took the

median time for each. I started with $N = 96$, increasing by eights up to to $N = 192$, for a total of 130 tests. The results of this process are shown in the right hand columns of Table 5.15.

Table 5.15: Scaling with respect to N

	$W = 2^{10}$		$W = 2^{11}$	
N	Time (sec)	Store (MB)	Time (sec)	Store (MB)
96	5.497	14.99	16.904	31.86
104	7.673	15.54	23.290	32.20
112	7.049	16.35	20.157	35.36
120	8.871	17.35	26.717	33.51
128	6.938	17.25	22.365	33.91
136	12.190	18.24	36.601	35.05
144	10.982	17.99	29.952	36.49
152	13.139	18.54	36.703	41.27
160	11.420	18.66	34.801	40.27
168	15.727	20.97	44.959	40.66
176	15.332	21.38	44.873	43.06
184	18.243	21.38	53.689	43.20
192	12.796	21.60	40.029	43.66

The scaling with respect to N, with W fixed at 2^{11}, is approximated by the equations shown on the right side of Table 5.16.

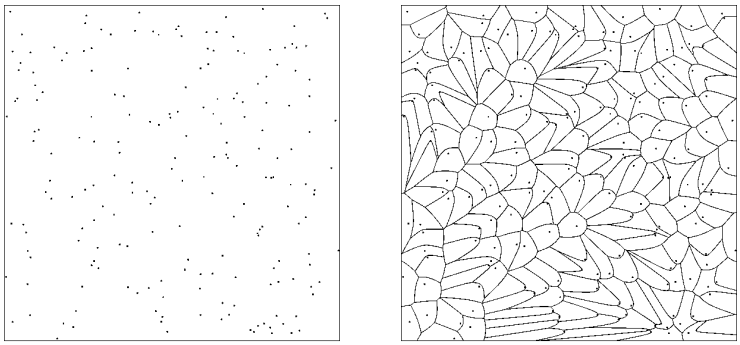
Table 5.16: Time and memory scaling with respect to N alone

Time	$T(N) \approx 4.582 \times 10^{-2} N \ln N$	$R^2 = 0.981$
Storage	$S(N) \approx 3.162\, N^{1/2}$	$R^2 = 0.999$

5.5.5.3 Scaling interaction on the plane: The effectiveness of the boundary method for large N

Increasing N means one must consider the scaling behavior of the boundary method with respect to N, as described in Section 5.5.5.2, above. However, there is another relevant limiting factor for $N \gg 1$: the decreasing area size $\mu(A_i) = n^{-1}$ runs up against the accuracy of the reconstruction. For the problem shown in Figure 5.11, the area of each

region is 5.0×10^{-3}. When $w_* = 2^{-11}$, my standard discretization, the maximum error for the area of the partition regions is 8.34×10^{-4}. This is an error of 16.7%, approximately one-sixth of the size of the affected region.



(a) $N = 200$ points in $\mathbb{R}^2$ (b) μ-partition for $N = 200$

Figure 5.11: Partitioning with large N

If all one desires is the Wasserstein distance or the boundary set, this reconstruction error need not be a concern. However, if one wants the set of shifts to generate an accurate reconstruction, large N requires that one increase W to match. Hence, it is necessary to consider what happens as W and N increase in tandem.

First of all, I require a minimum starting discretization which is dependent on N. Applying the minimum w_1 defined with Equations (5.4.1) and (5.4.2), and taking $W = 1/w_*$ gives $\min W = 50N$, said minimum being achieved only when N is a power of 2. Hence, all the test results shown in Table 5.15 were run with $w_1 = 2^{-7}$, which means four iterations were performed to get $W = 2^{10}$ and five for $W = 2^{11}$.

By comparison, the Barrett and Prigozhin example problem starts with $w_1 = 2^{-4}$, so when $W = 2^{11}$ the solution is the result of eight iterations. In order to maintain a constant number of iterations, one would want to first consider the interaction between W and N with respect to scaling. This can be done by considering the results in Table 5.15 with respect to the changes in both W and N. As it turns out, the scaling behavior one observes is consistent with the product of the two scaling behaviors already determined:

$\mathcal{O}(WN \log W \log N)$ with respect to time, and $\mathcal{O}(WN^{1/2})$ with respect to storage. See Table 5.17 for approximate equations.

Table 5.17: Time and memory scaling with respect to both W and N

Time	$T(N,\, W) \approx 2.853 \times 10^{-6} WN \ln W \ln N$	$R^2 = 0.971$
Storage	$S(N,\, W) \approx 1.538 \times 10^{-3}\, WN^{1/2}$	$R^2 = 0.999$

Suppose one wishes to complete a fixed R iterations on the plane, dividing the regions in two with each iteration. If

$$W_1^2 = 1/w_1^2 = 50N,$$

then

$$W = W_R = 2^R W_1 = 2^R (50N)^{1/2}.$$

Hence, since N is constant, the scaling behavior with respect to N can be computed as

$$\begin{aligned} T(N) &= T(N,\, 2^R(50N)^{1/2}) \\ &\approx 2.853 \times 10^{-6} (2^R(50N)^{1/2}) N \ln(2^R(50N)^{1/2}) \ln N \sim \mathcal{O}(N^{3/2}(\log N)^2) \end{aligned}$$

and

$$S(N) = S\left(N,\, 2^R(50N)^{1/2}\right) \approx 1.538 \times 10^{-3}\, \left(2^R(50N)^{1/2}\right) N^{1/2} \sim \mathcal{O}(N).$$

5.5.5.4 Scaling in three dimensions and extrapolation to $\mathbb{R}^d$

The computations described above can be repeated in three dimensions. I first scale W separately by taking a projection of the Barrett and Prigozhin example into the center of the cube $[0,\, 1]^3$. Then I examine N by considering the median of ten tests when $W = 2^7$ and N starts at 8 and ends at 80, incremented by 8 each time. Finally, I bring the two results together, and consider the combined bounds.

The results for independent scaling of W and N are given in Table 5.18 and the rightmost columns of Table 5.19, respectively. Combined scaling data is presented in the other columns of Table 5.19. Approximate equations for three dimensional scaling are presented in Table 5.20.

Table 5.18: Scaling with respect to W

W	Time (sec)	Store (MB)
2^3	0.003	0.483
2^4	0.027	3.533
2^5	0.147	19.240
2^6	0.869	86.910
2^7	4.482	368.200
2^8	22.283	1514.496
2^9	113.129	6141.952

Table 5.19: 3-D scaling with respect to N

	$W = 2^4$		$W = 2^5$		$W = 2^6$		$W = 2^7$	
N	T (sec)	S (MB)	T (sec)	S (MB)	T (sec)	S (MB)	T (sec)	S (MB)
8	0.029	3.690	0.212	22.930	1.248	110.800	7.208	494.900
16	0.036	3.730	0.313	25.450	2.306	130.900	13.861	579.200
24	0.054	3.884	0.459	27.290	3.712	148.700	22.068	673.400
32	0.062	3.766	0.517	28.380	4.521	158.700	31.181	731.900
40	0.088	3.955	0.742	29.220	6.798	171.100	44.601	801.700
48	0.104	3.973	0.804	29.260	7.974	178.000	53.369	844.600
56	0.121	4.030	1.051	29.520	11.428	182.800	71.438	873.700
64	0.098	3.864	0.976	29.480	9.511	186.800	69.546	907.700
72	0.153	4.282	1.422	30.660	14.029	195.100	93.699	984.200
80	0.164	4.175	1.413	30.650	13.655	194.600	93.282	979.400

Suppose one wishes to complete a fixed R iterations in three dimensions. Then

$$W_1^3 = 1/w_1^3 = 50N$$

Table 5.20: Time and memory scaling with respect to both W and N

W alone	Time	$T(W) \approx 6.878 \times 10^{-5} W^2 \ln W$	$R^2 = 0.999$
	Storage	$S(W) \approx 2.341 \times 10^{-2}\, W^2$	$R^2 = 1.000$
N alone	Time	$T(N) \approx 2.849 \times 10^{-1} N \ln N$	$R^2 = 0.995$
	Storage	$S(N) \approx 2.315 \times 10^{2}\, N^{1/3}$	$R^2 = 1.000$
W and N	Time	$T(N,\, W) \approx 3.531 \times 10^{-6} W^2 N \ln W \ln N$	$R^2 = 0.989$
	Storage	$S(N,\, W) \approx 1.397 \times 10^{-1}\, W^2 N^{1/3}$	$R^2 = 0.999$

and

$$W = W_R = 2^R W_1 = 2^R (50N)^{1/3}.$$

This gives scaling behavior with respect to N as

$$\begin{aligned} T(N) &= T(N,\, 2^R(50N)^{1/3}) \\ &\approx 3.531 \times 10^{-6} (2^R(50N)^{2/3}) N \ln(2^R(50N)^{1/3}) \ln N \sim \mathcal{O}(N^{5/3}(\log N)^2) \end{aligned}$$

and

$$S(N) = S\left(N,\, 2^R(50N)^{1/3}\right) \approx 1.397 \times 10^{-1}\, \left(2^R(50N)^{2/3}\right) N^{1/3} \sim \mathcal{O}(N).$$

Taking the combined scaling equations for two and three dimensions, and extrapolating to arbitrary dimension $d \geq 2$, I anticipate average scaling of

$$T(d,\, N,\, W) \sim \mathcal{O}(W^{d-1} N \log W \log N) \qquad \text{and} \qquad S(d,\, N,\, W) \sim \mathcal{O}(W^{d-1} N^{1/d}). \tag{5.5.14}$$

When $W_1 = (50N)^{1/d}$ and one wishes to complete a fixed R iterations, this gives

$$T(d,\, N) \sim \mathcal{O}\left(N^{2-1/d} (\log N)^2\right) \qquad \text{and} \qquad S(d,\, N) \sim \mathcal{O}(N). \tag{5.5.15}$$

CHAPTER 6
CONCLUSION

I built efficient new algorithms for numerical optimal transport. I showed a new technique for discrete transport problems, the general auction, and used it as the basis for a semi-discrete solver, the boundary method.

When applied to discrete and semi-discrete optimal transport, the general auction and boundary method satisfy the objectives (a), (b), and (c) described in the Introduction. Both methods are able to handle general ground costs, they efficiently compute Wasserstein distances, and they have already been applied successfully to both two and three-dimensional problems.

Furthermore, the algorithms already form the kernel of a new continuous optimal transport solver. As written, the boundary method constitutes a method of mesh generation, which is sufficient for most numerical continuous transport applications. If the discretization of Y is known to be grid-based, the method can be optimized even further.

That said, important mathematical development still needs to be done in order to fully realize the potential of my work. As described in Chapter 5, the boundary method assumes that all integrals and all computations are performed exactly. To show robustness, it is important to consider what happens when those values are merely approximations. While *ad hoc* experiments indicate that the method is quite forgiving of approximated values, the extent of this robustness needs to be formalized and quantified.

When considering applications of my research to continuous transport, even more work needs to be done. For example, a variant boundary algorithm would be more effective on continuous problems if X and Y were refined in tandem. I have already begun testing this approach, but there are important mathematical issues to resolve. The solution to a fully continuous transport problem may not be unique, even if the solution of its semi-discrete

approximation is. Also, I have not yet shown the mathematical connection between the solutions of two semi-discrete problems generated from the same continuous source.

While I am not yet ready to guarantee that the transport maps or the Wasserstein distances converge when approximating continuous transport solutions, I believe this is a worthwhile direction for further research. In this respect, there may be a great advantage to following up further on Remark 5.3.30. Every result from the boundary method solves some optimal transport problem, and the problem solved must be related to the problem whose solution is desired. This could be the basis for a backward error analysis that shifts responsibility to some underlying conditioning of the problem.

In the meantime, there are already many promising applications for the semi-discrete boundary method. Transport is a type of optimization problem which generalizes exceedingly well. For instance, one can use the boundary method to subdivide a continuous or discrete space into appropriately-designed regions. These could be zones of control for national defense, delivery areas for shipping, congressional districts for federal apportionment, or many other possibilities.

The boundary method can also make important contributions to ongoing research in economics and machine learning, as it already provides the basis of a robust clustering algorithm. Unlike most existing methods of clustering, mass transport naturally balances consideration of centroids, distributions, and densities. Its integral definition also provides innate robustness with respect to outliers.

Because of its fine scaling properties, the boundary method could be used whenever a generalized Voronoi diagram is needed. The boundary method offers an alternative method of Voronoi cell generation, one that combines distance control with constraints on region volumes.

Other applications follow from the addition of an artificial time variable to the boundary method, such that for all $i \in \mathbb{N}_n$, $\mathbf{y}_i := \mathbf{y}_i(t)$ for some fixed t. Numerical experiments suggest that most facets of the semi-discrete problem change smoothly with respect to t.

Introducing this, one could add time-based variation to existing applications. For example, by constructing zones of control that change with respect to time, one could greatly increase their practical usefulness.

I have only begun to tap the potential of these numerical transport methods. I look forward to discovering what the future brings.

Appendices

APPENDIX A

MONGE UNDER THE SHIFT CHARACTERIZATION

A.1 ℓ_p functions with $p \in (1, \infty)$

If $c = \ell_p$ with $p \in (1, \infty)$, then the semi-discrete transport problem is always Monge under the shift characterization. This is shown in two steps:

(1) If the function g_{ij} is equal to the constant value $a_i - a_j$ in some neighborhood of $\mathbf{x}_0 \in A_{ij}$, then $|a_i - a_j| = c(\mathbf{y}_i, \mathbf{y}_j)$. [Theorem A.1.1]

(2) It follows from Step (1) that $\mu(B) > 0$ implies the existence of a ball of positive radius whose points are all collinear with both $\mathbf{y}_i$ and $\mathbf{y}_j$. [Theorem A.1.2]

Because of the contradiction inherent in Step (2), $\mu(B) = 0$, and so Theorem A.1.2 concludes that the problem must be Monge under the shift characterization.

Theorem A.1.1. *Let c be an ℓ_p norm with $p \in (1, \infty)$, and $\mathbf{x}_0 \in A_{ij}$ for some $i, j \in \mathbb{N}_n$, $i \neq j$. If $g_{ij}(\mathbf{x}) = a_i - a_j$ for all $\mathbf{x}$ in a neighborhood of $\mathbf{x}_0$, then $|a_i - a_j| = c(\mathbf{y}_i, \mathbf{y}_j)$.*

Proof. Let c be an ℓ_p norm with $p \in (1, \infty)$, $\mathbf{x}_0 \in A_{ij}$, and $g_{ij}(\mathbf{x}) = a_i - a_j$ for all $\mathbf{x}$ in some neighborhood of $\mathbf{x}_0$. Suppose to the contrary, however, that $|a_i - a_j| \neq c(\mathbf{y}_i, \mathbf{y}_j)$.

Say $|a_i - a_j| > c(\mathbf{y}_i, \mathbf{y}_j)$, and assume without loss of generality that $|a_i - a_j| = a_i - a_j$. Then

$$g_{ij}(\mathbf{x}_0) = c(\mathbf{x}_0\, \mathbf{y}_i) - c(\mathbf{x}_0\, \mathbf{y}_j) = a_i - a_j > c(\mathbf{y}_i, \mathbf{y}_j),$$

which implies $c(\mathbf{x}_0\, \mathbf{y}_i) > c(\mathbf{x}_0\, \mathbf{y}_j) + c(\mathbf{y}_i, \mathbf{y}_j)$. This is a violation of the triangle inequality. Therefore, it must be the case that $|a_i - a_j| < c(\mathbf{y}_i, \mathbf{y}_j)$.

For all $k \in \mathbb{N}_n$, define $c_k(\mathbf{x}) := c(\mathbf{x}, \mathbf{y}_k)$. Because $|a_i - a_j| < c(\mathbf{y}_i, \mathbf{y}_j)$, $\mathbf{x}_0 \neq \mathbf{y}_i$ and $\mathbf{x}_0 \neq \mathbf{y}_j$. Hence, $c_i(\mathbf{x}_0) > 0$ and $c_j(\mathbf{x}_0) > 0$.

Because g_{ij} is constant in a neighborhood of $\mathbf{x}_0$,

$$\nabla g_{ij}(\mathbf{x}_0) = \nabla c_i(\mathbf{x}_0) - \nabla c_j(\mathbf{x}_0) = 0,$$

which implies $\nabla c_i(\mathbf{x}_0) = \nabla c_j(\mathbf{x}_0)$. Hence, each of the first-order partial derivatives of c_i and c_j are equal at $\mathbf{x}_0$.

Assume $\mathbf{x}_0 = (x_1, \ldots, x_d)$, $\mathbf{y}_i = (y_1^i, \ldots, y_d^i)$, and $\mathbf{y}_j = (y_1^j, \ldots, y_d^j)$. Then the equality of the k-th partial derivatives, $\nabla_{x_k} c_i(\mathbf{x}_0) = \nabla_{x_k} c_j(\mathbf{x}_0)$, gives

$$(x_k - y_k^i)|x_k - y_k^i|^{p-2} \left(c_i(\mathbf{x}_0)\right)^{1-p} = (x_k - y_k^j)|x_k - y_k^j|^{p-2} \left(c_j(\mathbf{x}_0)\right)^{1-p}.$$

Thus, $x_k - y_k^i$ and $x_k - y_k^j$ have the same sign or are both zero. Because $p > 1$, $p - 1 > 0$. Hence, taking the $(p-1)$-th root of both sides,

$$\frac{x_k - y_k^i}{c_i(\mathbf{x}_0)} = \frac{x_k - y_k^j}{c_j(\mathbf{x}_0)} \qquad \forall\, k \in \mathbb{N}_d. \tag{A.1.1}$$

As a consequence of Equation (A.1.1), $x_k - y_k^i = 0$ if and only if $x_k - y_k^j = 0$. Hence, $x_k = y_k^i$ if and only if $x_k = y_k^j$.

Let K be the total number of k-th directional components satisfying $x_k \neq y_k^i$. Consider three cases: $K = 0$, $K = 1$, and $K > 1$.

$K = 0$: Then $x_k = y_k^i = y_k^j$ for all $k \in \mathbb{N}_d$, in which case $\mathbf{y}_i = \mathbf{y}_j$. Since the semi-discrete transport problem requires distinct non-zero points in Y, it must be the case that $i = j$, contradicting the initial assumption that $i \neq j$. Hence, $K \geq 1$.

$K = 1$: There exists exactly one k such that the components are not equal. Since $x_k - y_k^i$ and $x_k - y_k^j$ have the same sign,

$$|g_{ij}(\mathbf{x}_0)| = |(x_k - y_k^i) - (x_k - y_k^j)| = |y_k^j - y_k^i| = c(\mathbf{y}_i, \mathbf{y}_j).$$

This contradicts the assumption that $|a_i - a_j| < c(\mathbf{y}_i, \mathbf{y}_j)$, and hence $K > 1$.

$K > 1$: Because g_{ij} is constant in some neighborhood of $\mathbf{x}_0$, it must also be the case that $\nabla^2 g_{ij}(\mathbf{x}_0) = 0$. Hence, $\nabla^2 c_i(\mathbf{x}_0) = \nabla^2 c_j(\mathbf{x}_0)$, so each of the second-order partial derivatives of c_i and c_j are equal at $\mathbf{x}_0$.

The equality of the second-order partial derivatives taken with respect to x_k gives

$$\frac{(p-1)|x_k - y_k^i|^{p-2}}{(c_i(\mathbf{x}_0))^{2p-1}} \left[(c_i(\mathbf{x}_0))^p - |x_k - y_k^i|^p\right] = \frac{(p-1)|x_k - y_k^j|^{p-2}}{(c_j(\mathbf{x}_0))^{2p-1}} \left[(c_j(\mathbf{x}_0))^p - |x_k - y_k^j|^p\right], \quad \text{(A.1.2)}$$

which can be rewritten as

$$\frac{p-1}{c_i(\mathbf{x}_0)} \left(\frac{|x_k - y_k^i|}{c_i(\mathbf{x}_0)}\right)^{p-2} \left[1 - \left(\frac{|x_k - y_k^i|}{c_i(\mathbf{x}_0)}\right)^p\right] = \frac{p-1}{c_j(\mathbf{x}_0)} \left(\frac{|x_k - y_k^j|}{c_j(\mathbf{x}_0)}\right)^{p-2} \left[1 - \left(\frac{|x_k - y_k^j|}{c_j(\mathbf{x}_0)}\right)^p\right]. \quad \text{(A.1.3)}$$

Applying Equation (A.1.1), define

$$\sigma_k = \frac{|x_k - y_k^i|}{c_i(\mathbf{x}_0)} = \frac{|x_k - y_k^j|}{c_j(\mathbf{x}_0)}.$$

Then Equation (A.1.3) can be rewritten as

$$\frac{p-1}{c_i(\mathbf{x}_0)} \sigma_k^{p-2}(1 - \sigma_k^p) = \frac{p-1}{c_j(\mathbf{x}_0)} \sigma_k^{p-2}(1 - \sigma_k^p). \quad \text{(A.1.4)}$$

By assumption, for all $k \in \mathbb{N}_d$, $x_k - y_k^i \neq 0$ and $x_k - y_k^j \neq 0$. Hence, $\sigma_k > 0$.

Since $d > 1$, and $|x_k - y_k^i| > 0$ for all $k \in \mathbb{N}_d$, it must be that $|x_k - y_k^i| < c_i(\mathbf{x}_0)$ for all $k \in \mathbb{N}_d$. Therefore,

$$\sigma_k = \frac{|x_k - y_k^i|}{c_i(\mathbf{x}_0)} < 1,$$

which implies

$$1 - \sigma_k^p > 0.$$

Therefore, $(p-1)\,\sigma_k^{p-2}(1-\sigma_k^p) > 0$, and Equation (A.1.4) simplifies to

$$\frac{1}{c_i(\mathbf{x}_0)} = \frac{1}{c_j(\mathbf{x}_0)}.$$

Thus, $c_i(\mathbf{x}_0) = c_j(\mathbf{x}_0)$. Combining this with Equation (A.1.1) implies $y_k^i = y_k^j$ for all $k \in \mathbb{N}_d$, and so $\mathbf{y}_i = \mathbf{y}_j$.

Since $\mathbf{y}_i = \mathbf{y}_j$, and the semi-discrete transport problem requires distinct non-zero points in Y, it must be the case that $i = j$, contradicting the initial assumption that $i \neq j$. Thus, $K \not> 1$.

All choices of K lead to contradictions. Hence, if $c = \ell_p$ for some $p \in (1, \infty)$, $\mathbf{x}_0 \in A_{ij}$, $i \neq j$, and $g_{ij}(\mathbf{x}) = a_i - a_j$ for all $\mathbf{x}$ in some neighborhood of $\mathbf{x}_0 \in A_{ij}$, then it must be the case that $|a_i - a_j| = c(\mathbf{y}_i, \mathbf{y}_j)$. □

Theorem A.1.2. *If $c = \ell_p$ for some $p \in (1, \infty)$, then the semi-discrete transport problem is Monge under the shift characterization.*

Proof. Assume the contrary is true. Then $\mu(B) > 0$, so $\mu(A_{ij}) > 0$ for some $i, j \in \mathbb{N}_n$, $i \neq j$. Because μ is nonatomic, there exist $\mathbf{x}_0 \in A_{ij}$ and $\epsilon > 0$ such that the ball $\mathcal{B}_\epsilon(\mathbf{x}_0)$, defined with respect to the Euclidean space $\mathbb{R}^d$, satisfies $\mathcal{B}_\epsilon(\mathbf{x}_0) \subseteq A_{ij}$ and $\mu(\mathcal{B}_\epsilon(\mathbf{x}_0)) > 0$. By Theorem A.1.1, $|a_i - a_j| = c(\mathbf{y}_i, \mathbf{y}_j)$. Assume without loss of generality that $|a_i - a_j| = a_i - a_j$.

Let $\mathbf{x} \in \mathcal{B}_\epsilon(\mathbf{x}_0)$. Since $\mathbf{x} \in A_{ij}$,

$$\begin{aligned} g_{ij}(\mathbf{x}) &= a_i - a_j \\ c(\mathbf{x}, \mathbf{y}_i) - c(\mathbf{x}, \mathbf{y}_j) &= c(\mathbf{y}_i, \mathbf{y}_j) \\ c(\mathbf{x}, \mathbf{y}_i) &= c(\mathbf{x}, \mathbf{y}_j) + c(\mathbf{y}_i, \mathbf{y}_j) \end{aligned}$$

Because $c = \ell_p$ and $p \in (1, \infty)$, Minkowski's inequality implies that $\mathbf{x}$, $\mathbf{y}_i$, and $\mathbf{y}_j$ are all collinear. The choice of $\mathbf{x}$ was nonspecific, and therefore every point in the ball $\mathcal{B}_\epsilon(\mathbf{x}_0)$ must be collinear with the points $\mathbf{y}_i$ and $\mathbf{y}_j$.

Of course, this is impossible, and so $\mu(A_{ij}) = 0$ for all $i, j \in \mathbb{N}_n$, $i \neq j$. Therefore, $\mu(B) = 0$. From this final contradiction, it is clear that the semi-discrete transport problem must be Monge under the shift characterization. □

REFERENCES

[1] L. Ambrosio and A. Pratelli, "Existence and stability results in the l^1 theory of optimal transportation," in *Optimal Transportation and Applications: Lectures given at the C.I.M.E. Summer School held in Martina Franca, Italy, September 2–8, 2001*, ser. Lecture Notes in Mathematics, vol. 1813, Springer-Verlag, 2004, pp. 123–160.

[2] A. Andoni, A. Nikolov, K. Onak, and G. Yaroslavtsev, "Parallel algorithms for geometric graph problems," in *STOC '14 Proceedings of the forty-sixth annual ACM symposium on Theory of Computing: New York, New York — May 31 – June 03, 2014*, ser. Contemporary Mathematics, vol. 226, New York: ACM, 2014, pp. 574–583.

[3] S. Angenent, S. Haker, and A. Tannenbaum, "Minimizing flows for the Monge-Kantorovich problem," *SIAM J. Math. Anal.*, vol. 35, no. 1, pp. 61–97, 2003.

[4] S. Barr, J. Wang, and B. Liu, "An efficient method for constructing underwater sensor barriers," *Journal of Communications*, vol. 6, no. 5, pp. 370–383, 2011.

[5] J. W. Barrett and L. Prigozhin, "A mixed formulation of the Monge-Kantorovich equations," *ESAIM: M2AN*, vol. 41, pp. 1041–1060, 6 2007.

[6] M. Beiglböck, P. Henry-Labordére, and F. Penkner, "Model-independent bounds for option prices: a mass transport approach," *Finance and Stochastics*, vol. 17, pp. 477–501, 3 2013.

[7] J.-D. Benamou, "Numerical resolution of an "unbalanced" mass transport problem," *ESAIM: Mathematical Modeling and Numerical Analysis*, vol. 37, no. 5, pp. 851–868, 2003.

[8] J.-D. Benamou and Y. Brenier, "A numerical method for the optimal time-continuous mass transport problem and related problems," in *Monge Ampére equation: applications to geometry and optimization (NSF-CBMS Conference on the Monge Ampére Equation, Applications to Geometry and Optimization, July 9–13, 1997, Florida Atlantic University)*, ser. Contemporary Mathematics, vol. 226, Providence, R.I.: American Mathematical Society, 1999, pp. 1–11.

[9] J.-D. Benamou, Y. Brenier, and K. Guittet, "The Monge-Kantorovich mass transfer and its computational fluid mechanics formulation," *International Journal for Numerical Methods in Fluids*, vol. 40, no. 1–2, pp. 21–30, 2000.

[10] J.-D. Benamou and G. Carlier, "Augmented Lagrangian methods for transport optimization, mean-field games and degenerate PDEs," *Journal of Optimization Theory and Applications*, vol. 167, no. 1, pp. 1–26, 2015.

[11] J.-D. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré, "Iterative Bregman projections for regularized transportation problems," *SIAM Journal on Scientific Computing*, vol. 37, no. 2, A1111–A1138, 2015.

[12] J.-D. Benamou, G. Carlier, and L. Nenna, "A numerical method to solve optimal transport problems with Coulomb cost," arXiv:1505.01136v2, 2015.

[13] J.-D. Benamou, B. D. Froese, and A. M. Oberman, "Two numerical methods for the elliptic Monge-Ampère equation," *ESAIM: Mathematical Modeling and Numerical Analysis*, vol. 44, no. 4, pp. 737–758, 2009.

[14] ——, "Numerical solution of the optimal transportation problem using the Monge-Ampère equation," *Journal of Computational Physics*, vol. 260, pp. 107–126, 2014.

[15] D. P. Bertsekas, *Network Optimization: Continuous and Discrete Models*. Belmont, Massachusetts: Athena Scientific, 1998, `http://web.mit.edu/dimitrib/www/books.htm`. Accessed: 2016-09-16.

[16] D. P. Bertsekas and D. A. Castañón, "The auction algorithm for the transportation problem," *Annals of Operations Research*, vol. 20, no. 1, pp. 67–96, 1989.

[17] ——, "A generic auction algorithm for the minimum cost network flow problem," *Comput. Optim. Appl.*, vol. 2, pp. 229–260, 1993.

[18] D. P. Bertsekas and P. Tseng, "Relaxation methods for minimum cost ordinary and generalized network flow problems," *Operations Research*, vol. 36, no. 1, pp. 93–114, 1988.

[19] A. Blanchet and G. Carlier, "Optimal transport and Cournot-Nash equilibria," arXiv:1206.6571, 2012.

[20] R. G. Bland and D. L. Jensen, "On the computational behavior of a polynomial-time network flow algorithm," *Mathematical Programming*, vol. 54, no. 1–3, pp. 1–39, 1992.

[21] D. Bosc, *Numerical approximation of optimal transport maps*, 2010.

[22] G. Bouchitté, G. Buttazzo, and P. Seppecher, "Shape optimization solutions via Monge-Kantorovich equation," *Comptes Rendus de l'Académie des Sciences – Series I – Mathematics*, vol. 324, pp. 1185–1191, 10 1997.

[23] A. Bouharguane, A. Iollo, and L. Weynans, "Numerical solution of the Monge-Kantorovich problem by Picard iterations," Inria Research Centre, Tech. Rep., 2014.

[24] Y. Brenier, "Décomposition polaire et réarrangement monotone des champs de vecteurs," *Comptes Rendus de l Académie des Sciences, Série I*, vol. 305, pp. 805–808, 1987.

[25] C. J. Budd and J. F. Williams, "Moving mesh generation using the parabolic Monge-Ampère equation," *SIAM Journal on Scientific Computing*, vol. 31, no. 5, pp. 3438–3465, 2009.

[26] G. Buttazzo, "Three optimization problems in mass transportation theory," in *Non-smooth Mechanics and Analysis: Theoretical and Numerical Advances*, ser. Advances in Mechanics and Mathematics, vol. 12, Springer, 2006, pp. 13–23.

[27] G. Buttazzo, L. De Pascale, and P. Gori-Giorgi, "Optimal-transport formulation of electronic density-functional theory," *Physical Review A*, vol. 85, no. 062502, 2012.

[28] L. A. Caffarelli, S. A. Kochengin, and V. I. Oliker, "On the numerical solution of the problem of reflector design with given far-field scattering data," in *Monge Ampére equation: applications to geometry and optimization (NSF-CBMS Conference on the Monge Ampére Equation, Applications to Geometry and Optimization, July 9–13, 1997, Florida Atlantic University)*, ser. Contemporary Mathematics, vol. 226, Providence, R.I.: American Mathematical Society, 1999, pp. 13–32.

[29] L. A. Caffarelli and V. I. Oliker, "Weak solutions of one inverse problem in geometric optics," *Journal of Mathematical Sciences*, vol. 154, no. 1, pp. 39–49, 2008.

[30] E. A. Carlen and W. Gangbo, "Solution of a model Boltzmann equation via steepest descent in the 2-Wasserstein metric," *Archive for Rational Mechanics and Analysis*, vol. 172, pp. 21–64, 1 2004.

[31] G. Carlier, "A general existence result for the principal-agent problem with adverse selection," *Journal of Mathematical Economics*, vol. 35, pp. 129–150, 1 2001.

[32] ——, "Optimal transportation and economic applications," 2010, IMA, New Mathematical Models in Economics and Finance. Lecture notes.

[33] G. Carlier and I. Ekeland, "Matching for teams," *Economic Theory*, vol. 42, pp. 397–418, 2 2010.

[34] G. Carlier and F. Santambrogio, "A continuous theory of traffic congestion and Wardrop equilibria," *Journal of Mathematical Sciences*, vol. 181, pp. 792–804, 6 2012.

[35] R. Chartrand, B. Wohlberg, K. R. Vixie, and E. M. Bollt, "A gradient descent solution to the Monge-Kantorovich problem," *Applied Mathematical Sciences (Ruse)*, vol. 3, pp. 1071–1080, 21–24 2009.

[36] P.-A. Chiappori, R. McCann, and B. Pass, "Multi- to one-dimensional optimal transport," To appear in Comm. Pure Appl. Math, 2016.

[37] J.-F. Cossette and P. K. Smolarkiewicz, "A Monge-Ampére enhancement for semi-Lagrangian methods," *Computers & Fluids*, vol. 46, pp. 180–185, 1 2011.

[38] J. A. Cuesta-Albertos and A. Tuero-Díaz, "A characterization for the solution of the Monge-Kantorovich mass transference problem," *Statistics and Probability Letters*, vol. 16, no. 2, pp. 147–152, 1993.

[39] M. J. P. Cullen, "A comparison of numerical solutions to the Eady frontogenesis problem," *Quarterly Journal of the Royal Meteorological Society*, vol. 134, pp. 2143–2155, 637 2008.

[40] M. J. P. Cullen and R. Douglas, "Applications of the Monge-Ampère equation and Monge transport problem to meteorology and oceanography," in *Monge Ampère Equation: Applications to Geometry and Optimization*, ser. Contemporary Mathematics, vol. 26, Providence, RI: American Mathematical Society, 1999, pp. 33–53.

[41] M. J. P. Cullen and R. J. Purser, "An extended Lagrangian theory of semi-geostrophic frontogenesis," *Journal of the Atmospheric Sciences*, vol. 41, pp. 1477–1497, 9 1984.

[42] M. Cuturi, "Sinkhorn distances: lightspeed computation of optimal transport," in *Advances in Neural Information Processing Systems*, vol. 26, Curran Associates, Inc., 2013, pp. 2292–2300.

[43] L.-P. S. Demers, "An efficient numerical algorithm for the L^2 optimal transport problem with applications to image processing," Master's thesis, University of Victoria, 2008.

[44] L. Dieci and J. Walsh III, "The boundary method for semi-discrete optimal transport partitions and Wasserstein distance computation," Under review, `http://arxiv.org/abs/1702.03517`, 2017.

[45] X. Dupuis, *The semi-discrete principal agent problem*, Presented at "Computational Optimal Transportation" workshop, July 18–22, 2016. `http://www.crm.umontreal.ca/2016/Optimal16/pdf/dupuis.pdf`.

[46] J. Edmonds and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems," *Journal of the Association for Computing Machinery*, vol. 19, pp. 248–264, 2 1972.

[47] U. Ehret and E. Zehe, "Series distance — an intuitive metric to quantify hydrograph similarity in terms of occurrence, amplitude and timing of hydrological events," *Hydrology and Earth System Sciences*, vol. 15, pp. 877–896, 2011.

[48] L. C. Evans, "Partial differential equations and Monge-Kantorovich mass transfer," in *Current developments in mathematics, 1997 (Cambridge, MA)*, Boston: International Press, 1999, pp. 65–126.

[49] L. C. Evans and W. Gangbo, *Differential equations methods for the Monge-Kantorovich mass transfer problem*, ser. Memoirs of the American Mathematical Society 653. Providence, R.I.: American Mathematical Society, 1999, vol. 137.

[50] M. Feldman, "Growth of a sandpile around an obstacle," in *Monge Ampére equation: applications to geometry and optimization (NSF-CBMS Conference on the Monge Ampére Equation, Applications to Geometry and Optimization, July 9–13, 1997, Florida Atlantic University)*, ser. Contemporary Mathematics, vol. 226, Providence, R.I.: American Mathematical Society, 1999, pp. 55–78.

[51] S. Ferradans, N. Papadakis, J. Rabin, G. Peyré, and J.-F. Aujol, "Regularized discrete optimal transport," in *Scale Space and Variational Methods in Computer Vision: 4th International conference, SSVM 2013, Schloss Seggau, Leibnitz, Austria, June 2013 Proceedings*, ser. Lecture Notes in Computer Science, vol. 7893, New York: Springer, 2013, pp. 428–439.

[52] S. Ferradans, G.-S. Xia, G. Peyré, and J.-F. Aujol, "Static and dynamic texture mixing using optimal transport," in *Scale Space and Variational Methods in Computer Vision: 4th International conference, SSVM 2013, Schloss Seggau, Leibnitz, Austria, June 2013 Proceedings*, ser. Lecture Notes in Computer Science, vol. 7893, New York: Springer, 2013, pp. 137–148.

[53] A. Figalli, Y.-H. Kim, and R. J. McCann, "When is multi-dimensional screening a convex program?" *Journal of Economic Theory*, vol. 146, pp. 454–478, 2 2011.

[54] J. H. Fitschen, F. Laus, and G. Steidl, *Dynamic optimal transport with mixed boundary condition for color image processing*, arXiv:1501.04840v1, 2015.

[55] L. R. Ford Jr. and D. R. Fulkerson, "Solving the transportation problem," *Management Science*, vol. 3, pp. 24–32, 1956.

[56] S. Fortune, "A sweepline algorithm for voronoi diagrams," in *Proceedings of the Second Annual Symposium on Computational Geometry*, ser. SCG '86, New York: ACM, 1986, pp. 313–322.

[57] U. Frisch, S. Matarrese, R. Mohayaee, and A. Sobolevski, "A reconstruction of the initial conditions of the universe by optimal mass transportation," *Nature*, vol. 417, no. 6886, pp. 260–262, 2002.

[58] B. Froese and A. Oberman, "Fast finite difference solvers for singular solutions of the elliptic MOnge-Ampère equation," *Journal of Computational Physics*, vol. 230, no. 3, pp. 818–834, 2011.

[59] D. R. Fulkerson, "An out-of-kilter method for minimal cost flow problems," *SIAM J. Appl. Math.*, vol. 9, pp. 18–27, 1961.

[60] A. Galichon, P. Henry-Labordére, and N. Touzi, "A stochastic control approach to no-arbitrage bounds given marginals, with an application to loopback options," *The Annals of Applied Probability*, vol. 24, pp. 312–336, 1 2014.

[61] A. Galichon and B. Salanié, "Matching with trade-offs: revealed preferences over competing characteristics," *Columbia University Department of Economics Discussion Paper Series*, no. 0910-14, 2010.

[62] ——, "Cupid's invisible hand: social surplus and identification in matching models," preprint, 2015.

[63] W. Gangbo and R. J. McCann, "The geometry of optimal transportation," *Acta Mathematica*, vol. 177, no. 2, pp. 113–161, 1996.

[64] A. V. Goldberg and R. E. Tarjan, "Finding minimum-cost circulations by canceling negative cycles," *Journal of the Association for Computing Machinery*, vol. 36, pp. 873–886, 4 1989.

[65] C. Gutiérrez, *The Monge-Ampère Equation*, ser. Progress in Nonlinear Differential Equations and Their Applications. Boston: Birkhäuser Science, 2001, vol. 44.

[66] E. Haber, T. Rehman, and A. Tannenbaum, "An efficient numerical method for the solution of the L_2 optimal mass transfer problem," *SIAM J. Sci. Comput.*, vol. 32, no. 1, pp. 197–211, 2010.

[67] S. Haker and A. Tannenbaum, "Optimal mass transport and image registration," in *IEEE Workshop on Variational and Level Set Methods in Computer Vision: Pro-*

ceedings: 13 July, 2001, Vancouver, Canada, Los Alamitos, California: IEEE Computer Society, 2001, pp. 29–36.

[68] A. Iollo and D. Lombardi, "Advection modes by optimal mass transfer," *Physical Review E*, vol. 89, no. 2, p. 022 923, 2014.

[69] M. Iri, "A new method for solving transportation-network problems," *Journal of the Operations Research Society of Japan*, vol. 3, no. 1, pp. 27–87, 1960.

[70] R. Jordan, D. Kinderlehrer, and F. Otto, "The variational formulation of the Fokker-Planck equation," *SIAM Journal on Mathematical Analysis*, vol. 29, no. 1, pp. 1–17, 1998.

[71] L. V. Kantorovich, "On the translocation of masses," *C.R. (Doklady) Acad. Sci. URSS (N.S.)*, vol. 37, pp. 199–201, 1942.

[72] ——, "On a problem of Monge," *Uspekhi Mat. Nauk*, vol. 3, pp. 225–226, 1948.

[73] M. Klein, "A primal method for minimal cost flow with applications to the assignment and transportation problems," *Management Science*, vol. 14, no. 3, pp. 205–220, 1967.

[74] P. Kovács, "Minimum-cost flow algorithms: an experimental evaluation," *Optimization Methods and Software*, vol. 30, no. 1, pp. 94–127, 2015.

[75] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[76] W. Li, S. Osher, and W. Gangbo, "Fast algorithms for Earth Mover's distance based on optimal transport and L_1 type regularization i," Preprint, 2016.

[77] H. Ling and K. Okada, "An efficient Earth Mover's distance algorithm for robust histogram comparison," *IEEE Transactions on Pattern Analysis and Machine Intelligence archive*, vol. 29, no. 5, pp. 840–853, 2007.

[78] L. Liu, D. A. Shell, and N. Michael, "From selfish auctioning to incentivized marketing," *Autonomous robotics*, vol. 37, no. 4, pp. 417–430, 2014.

[79] G. Loeper and F. Rapetti, "Numerical solution of the Monge-Ampère equation by a Newton's algorithm," *C.R. Acad. Sci. Paris. Ser. I*, vol. 340, no. 4, pp. 319–324, 2005.

[80] D. Lombardi, "Inverse problems for tumor growth modeling," PhD thesis, Institut de Mathématiques de Bordeaux, Sep. 2011.

[81] X. Ma, N. S. Trudinger, and X. Wang, "Regularity of potential functions of the optimal transportation function," *Arch. Ration. Mech. Anal.*, vol. 177, no. 2, pp. 151–183, 2005.

[82] R. Moeckel and B. Murray, "Measuring the distance between time series," *Physica D*, vol. 102, pp. 187–194, 3–4 1997.

[83] Mokaplan, *Inria international program: associate team proposal 2014–2016*, https://team.inria.fr/mokaplan/files/2014/09/MOKALIEN_Proposal_2013.pdf, 2013.

[84] ——, *Mokalien associate team report 2014 (1st year)*, https://team.inria.fr/mokaplan/files/2014/09/MOKALIEN_Report_2014.pdf, 2014.

[85] ——, *Mokalien associate team report 2015 (2nd year)*, https://team.inria.fr/mokaplan/files/2014/09/MOKALIEN_Report_2015.pdf, 2015.

[86] G. Monge, "Mémoire sur la théorie des déblais et des remblais," in *Histoire de l'Académie Royale des Sciences de Paris, avec les Mémoires de Mathématique et de Physique pour la même année*, In French, Académie des sciences (France)., 1781, pp. 666–704.

[87] C. Monma and M. Segal, "A primal algorithm for finding minimum-cost flows in capacitated networks with applications," *The Bell System Technical Journal*, vol. 61, pp. 949–968, 6 1982.

[88] M. Muskulus, "Distance-based analysis of dynamical systems and time series by optimal transport," https://www.math.leidenuniv.nl/scripties/muskulus-thesis.pdf. Accessed: 2016-09-16, PhD thesis, Universiteit Leiden, 2010.

[89] ——, "Distance-based methods for structural damage detection in offshore wind turbine installations," in *Proceedings of the 24th Congress on Condition Monitoring and Diagnostics Engineering Management*, M. Singh, R. B. K. N. Rao, and J. P. Liyanage, Eds., COMADEM International, COMADEM International, U.K., 2011, ISBN: 0954130723.

[90] M. Muskulus, S. Houweling, S. Verduyn-Lunel, and A. Daffertshofer, "Functional similarities and distance properties," *Journal of Neuroscience Methods*, vol. 183, no. 1, pp. 31–41, 2009.

[91] M. Muskulus, A. M. Slats, P. J. Sterk, and S. Verduyn-Lunel, "Fluctuations and determinism of respiratory impedance in asthma and chronic obstructive pulmonary disease," *Journal of Applied Physiology*, vol. 109, pp. 1582–1591, 2010.

[92] M. Muskulus and S. Verduyn-Lunel, "Wasserstein distances in the analysis of time series and dynamical systems," *Physica D*, vol. 240, no. 1, pp. 45–58, 2011.

[93] Netlib, *The Netlib repository at UTK and ORNL*, `http://www.netlib.org/lp/generators`, Accessed: 2016-07-03.

[94] A. Okabe, K. Sugihara, B. Boots, and S. N. Chiu, *Spatial Tessellations: concepts and applications of Voronoi diagrams*, 2nd ed., ser. Wiley Series in Probability and Statistics. New York: John Wiley & Sons, 2000.

[95] J. B. Orlin, "A polynomial time primal network simplex algorithm for minimum cost flows," *Mathematical Programming*, vol. 78, pp. 109–129, 2 1997.

[96] F. Otto, "The geometry of dissipative evolution equations: the porous medium equation," *Communications in Partial Differential Equations*, vol. 26, pp. 101–174, 1–2 2001.

[97] J. Rabin, G. Peyré, J. Delon, and M. Bernot, "Wasserstein barycenter and its applications to texture mixing," in *Scale Space and Variational Methods in Computer Vision: third international conference, SSVM 2011, Ein-Gedi, Israel, May 29–June 2, 2011*, ser. Lecture Notes in Computer Science, vol. 6667, Springer, 2011, pp. 435–446.

[98] S. Rachev and L. Rüschendorf, *Mass Transportation Problems. Vol I: Theory, Vol II: Applications*, ser. Probability and its applications. New York: Springer-Verlag, 1998.

[99] M. G. Resende and P. M. Pardalos, *Handbook of Optimization in Telecommunications*. New York: Springer, 2006.

[100] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[101] L. Rüschendorf, "Fréchet-bounds and their applications," in *Advances in Probability Distributions with Given Marginals*, G. Dall'Aglio, S. Kotz, and G. Salinetti, Eds., Amsterdam: Springer Netherlands, 1991, pp. 151–187.

[102] L. Rüschendorf, "Monge-Kantorovich transportation problem and optimal couplings," *Jahresbericht der Deutschen Mathematiker-Vereinigung*, vol. 109, no. 3, pp. 113–137, 2007.

[103] L. Rüschendorf and L. Uckelmann, "On optimal multivariate couplings," in *Distributions with given marginals and moment problems*, Kluwer Academic Publishers, 1997, pp. 261–273.

[104] ——, "Numerical and analytical results for the transportation problem of Monge-Kantorovich," *Metrika*, vol. 51, no. 3, pp. 245–258, 2000.

[105] E. G. Tabak and G. Trigila, "Data-driven optimal transport," *Communications on Pure and Applied Mathematics*, vol. 69, pp. 613–648, 4 2016.

[106] R. E. Tarjan, "Dynamic trees as search trees via Euler tours, applied to the network simplex algorithm," *Mathematical Programming*, vol. 78, no. 2, pp. 169–177, 1997.

[107] G. Tartavel, Y. Gousseau, and G. Peyré, "Constrained sparse texture synthesis," in *Scale Space and Variational Methods in Computer Vision: 4th International conference, SSVM 2013, Schloss Seggau, Leibnitz, Austria, June 2013 Proceedings*, ser. Lecture Notes in Computer Science, vol. 7893, New York: Springer, 2013, pp. 186–197.

[108] A. Undurti, "Planning under uncertainty and constraints for teams of autonomous agents," PhD thesis, Massachusetts Institute of Technology, 2011.

[109] C. Villani, *Topics in Optimal Transportation*, ser. Graduate Studies in Mathematics. Providence, R.I.: American Mathematical Society, 2003, vol. 58, ISBN: 9780821833124.

[110] ——, *Optimal Transport: Old and New*, ser. Grundlehren der mathematischen Wissenschaften. Berlin: Springer-Verlag, 2009, vol. 338.

[111] J. Walsh III, *The AUCTION ALGORITHMS IN C++ project*, Computer software, `http://gatech.jdwalsh03.com/coding.html`.

[112] J. Walsh III and L. Dieci, "General auction method for real-valued optimal transport," Under review, `http://gatech.jdwalsh03.com/index.html`, 2016.

[113] L. Zhu, Y. Yang, S. Haker, and A. Tannenbaum, "An image morphing technique based on optimal mass preserving mapping," *IEEE Transactions on Image Processing*, vol. 16, no. 6, pp. 1481–1495, 2007.

[114] L. Zhu, Y. Yang, A. Tannenbaum, and S. Haker, "Image morphing based on mutual information and optimal mass transport," in *Proceedings of the 2004 International Conference on Image Processing, ICIP 2004, Singapore, October 24-27, 2004*, 2004, pp. 1675–1678.

[115] E. Zuazua, *Control and design for fluids and structures*, Presented at "Paseo por la Geometría," Leioa, UPV/EHU. April 18. `http://paginaspersonales.deusto.es/enrique.zuazua/documentos_public/archivos/personal/conferencias/paseo-externalviewer.pdf`, 2012.

VITA

J.D. has led an interesting life. The first time he became homeless, he was seventeen. The Social Security Administration declared him "totally and permanently disabled" in 1997. Nonetheless, he raised his three children, taught himself computer programming, and attended college when his health permitted. In 2012, J.D. received his bachelor's degree with a double major in mathematics and philosophy, graduating summa cum laude from Western Michigan University. He was able to work regularly enough by 2013 that he no longer required disability support.

In 2014, the National Science Foundation awarded J.D. a graduate research fellowship for his ongoing research in numerical optimal transport. A year later they funded his semester-long collaboration with Michael Muskulus of the Norwegian University of Science and Technology. The ideas developed during that period form the basis of this work.

In middle school, J.D. taught himself 6502 Assembly Language in order to write a side-scrolling helicopter video game on a computer with 3.5KB of memory. He has been programming ever since. While he considers himself a dedicated amateur, J.D. has written code in over a dozen languages, he builds desktop computers, and he has been hired on multiple occasions to repair computer equipment, design websites, and maintain network servers.

J.D.'s upbringing and experiences have left him with a keen sense of social justice, and what happens in its absence. He is politically active, contributes his time to social causes, and volunteers at homeless shelters. His approach to activism focuses on teaching, raising people's awareness of issues and sharing knowledge they can use to better themselves.

In those rare moments when J.D. is not working, he reads voraciously and indulges his eclectic musical tastes. Somehow, he always finds time to referee tabletop role-playing games for his friends.